(2)

Technical Report 1426
June 1991

The Discrete Wavelet Transform

M. J. Shensa

Approved for public release; distribution is unlimited.

NAVAL OCEAN SYSTEMS CENTER
San Diego, California 92152-5000

J. D. FONTANA, CAPT, USN
Commander

R. T. SHEARER, Acting
Technical Director

ADMINISTRATIVE INFORMATION

This work was performed for the Office of the Chief of Naval Research, Independent Research Programs (IR), Arlington, VA 22217-5000, under program element 0601152N. The work was performed by Dr. Mark J. Shensa, a member of the Analysis Branch of the ASW Technology Division, Naval Ocean Systems Center, San Diego, CA 92152-5000.

Released by
M. B. Klausen, Head
Analysis Branch

Under authority of
R. H. Moore, Head
ASW Technology Division

ACKNOWLEDGMENTS

The author would especially like to thank Dzu K. Le for his meticulous reading of the manuscript and penetrating remarks. He is also grateful to David Stein for several stimulating discussions. Finally, he would like to thank an anonymous reviewer for greatly simplifying the proof of Theorem 5.2.

MA

SUMMARY

In a general sense, this report represents an effort to clarify the relationship of discrete and continuous wavelet transforms. More narrowly, it focuses on bringing together two separately motivated implementations of the wavelet transform, the *algorithme ˋa trous* and Mallat's multiresolution decomposition. It is observed that these algorithms are both special cases of a single filter bank structure, the discrete wavelet transform, the behavior of which is governed by one's choice of filters. In fact, the *ˋa trous* algorithm, originally devised as a computationally efficient implementation, is more properly viewed as a nonorthogonal multiresolution algorithm for which the discrete wavelet transform is exact. Moreover, it is shown that the commonly used Lagrange *ˋa trous* filters are in one-to-one correspondence with the convolutional squares of the Daubechies filters for orthonormal wavelets of compact support.

A systematic framework for the discrete wavelet transform is provided, and conditions are derived under which it computes the continuous wavelet transform exactly. Suitable filter contraints for finite energy and boundedness of the discrete transform are also derived. Finally, relevant signal-processing parameters are examined, and it is remarked that orthonormality is balanced by restrictions on resolution.

CONTENTS

APPENDICES

FIGURES

1. INTRODUCTION

Wavelets are rapidly finding application as a tool for the analysis of nonstationary signals. [1] – [5] However, with the notable exception of orthonormal wavelets [6] – [9], very little literature has been devoted to linking discrete implementations to the continuous transform. As in the case of the discrete Fourier transform, these implementations (or filter banks) can, and should, stand on their own as abstract decompositions of discrete time series. Their wide sweeping significance, however, lies in their interpretation as wavelet transforms. In a general sense, this report represents an effort to clarify the relationship of discrete and continuous wavelet transforms. More narrowly, it focuses on bringing together two separately motivated implementations of the wavelet transform. One of them, the *algorithme `a trous*[1] for nonorthogonal wavelets, [4] – [5], was developed for music synthesis [2] and is particularly suitable for signal processing. The other, the multiresolution approach of S. Mallat and Y. Meyer, originally used in image processing, employs orthonormal wavelets. [6] – [10] The latter algorithm, apart from its wavelet interpretation, was discovered previously in the form of quadrature mirror filter (QMF) filter banks with perfect reconstruction where it finds application in speech transmission and split-band coding. [11] – [13]

A glance at these two algorithms suffices to reveal closely related structures. In fact, apart from the constraints on their filters, the decimated *`a trous* [5] and Mallat algorithms are identical. We are thus led to examine the expanded family of algorithms encompassing both types of filters. In this vein, it is showr, that the Lagrange interpolation filters commonly employed by the *`a trous* algorithm are actually the squares (in a convolutional sense) of the Daubechies filters for compact orthonormal wavelets. We also derive conditions under which the discrete implementation computes a continuous wavelet transform exactly and find that they bear an intimate relationship to the *`a trous* constraints.

From a more general viewpoint the situation is as follows: The algorithms to be discussed all are filter bank structures (see figure 1.1). Their only distinguishing feature is the choice of two finite length filters, a lowpass filter **f** and a bandpass filter **g**. The lowpass condition, expressed more precisely as $\Sigma f_k = \sqrt{2}$, is necessary to the construction of a corresponding continuous wavelet function. The bandpass requirement, while apparently not essential to all applications, insures that finite energy signals lead to finite energy transforms (cf., section 6). Under these conditions the filter bank output will be referred to as the Discrete Wavelet Transform (DWT), a terminology which will become clear in the course of the report.

One class of DWT filter pairs are the Daubechies filters [8] which yield orthogonal wavelet decompositions and constitute, in more conventional terms, a QMF filter bank with perfect reconstruction. Another is that for which the lowpass filter satisfies the *`a trous* condition $f_{2k} = \delta(k) / \sqrt{2}$. Such filters, which simply serve to interpolate every other point, correspond to a nonorthogonal wavelet decomposition. As mentioned above, if they are further restricted to be Lagrangian interpolators, they

[1] Literally, "algorithm with holes", this terminology, taken from [5], refers to the fact that all the even coefficients of the relevant filter (with the exception of the center) are zero.

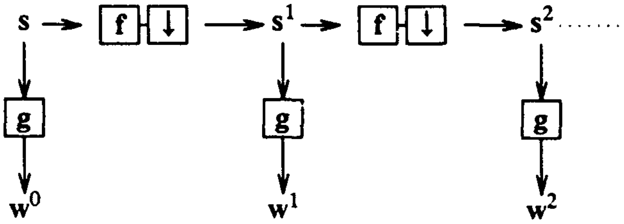

Figure 1.1 A wavelet filter bank structure. The down-arrow indicates decimation. The output of the transform is the family of signals $\mathbf{w}^i$, forming the two parameter transform $\mathbf{w}^i_n$ in the scale-time plane. Following terminology to be introduced, $\mathbf{w}^i$ is the (decimated) discrete wavelet transform.

become the squares of the Daubechies filters, which is quite remarkable in consideration of the totally different derivations. This also implies that a maximally flat filter with the same number of vanishing derivatives at 0 and π is a Lagrangian interpolator; a fact which may aid in the design of maximally flat filters. [14]

A fundamental question is when do these discrete implementations yield exact (i.e., sampled) versions of a continuous wavelet transform? Aside from regularity conditions relating to smoothness [8], we find that if **f** is ˋ*a trous*, then the DWT coincides with a continuous wavelet transform by a wavelet g(t) whose samples g(n) form the filter **g**. Even if **f** is not ˋ*a trous*, the algorithm is exact provided the signal lies in an appropriate subspace; however, in that instance, the corresponding wavelet depends on **f** as well as **g**. This is the situation in the orthonormal case where, moreover, the filter **g** is almost completely determined from **f** through the constraints of orthogonality.

In the remainder of this introduction we define, and briefly motivate, wavelet transforms at various levels of discretization. Section 2 contains an abbreviated derivation of the ˋ*a trous* algorithm followed by a description of the Mallat algorithm. (The uninitiated reader is particularly referred to [1], [6], [8].) In section 3 we define the undecimated DWT, relate it to the decimated transform, and provide algorithms for its computation. Section 4 states and proves several theorems which delineate the relationship between the DWT and the continuous wavelet transform. It may be read independently of the algorithms of section 2 although the motivation for the constructions may not be clear. Section 5 defines the Lagrange ˋ*a trous* filters and proves that they are the squares of the Daubechies filters. In section 6, we formulate the inversion problem and provide filter constraints which insure finite energy and bounded operators. It concludes with a short examination of the tradeoffs involved in choosing the bandpass filter, emphasizing the differences of the orthonormal and nonorthogonal cases.

TRANSFORM DEFINITIONS

The continuous wavelet transform of a signal s(t) takes the form

$$W(a,b) = \frac{1}{\sqrt{a}} \int \bar{g}(\frac{t-b}{a})\, s(t)\, dt \qquad (1.1)$$

where g is the analyzing wavelet, a represents a time dilation, b a time translation, and the bar stands for complex conjugate. The normalization factor $1/\sqrt{a}$ is perhaps most effectively visualized as endowing $| W(a,b) |^2$ with units of power/Hz.[2] Certain weak "admissibility" conditions are usually required on g(t) for it to be a candidate for an analyzing wavelet; namely, square integrability and

$$\int \frac{|\hat{g}(\omega)|^2}{|\omega|} d\omega < \infty \qquad (1.2)$$

where $\hat{g}(\omega)$ is the Fourier transform of g(t). They insure that the transformation is a bounded invertible operator in the appropriate spaces (cf., [8], [16]). If $\hat{g}(\omega)$ is differentiable, then it suffices that g be zero mean, i.e., that $\int g(t)dt = 0$, for equation (1.2) to be satisfied.

In the realm of signal processing, the significance of equation (1.1) is probably (or, at least, traditionally) best grasped by comparing it to the short-time Fourier transform:

$$F(\omega, b) = \int h(t-b)\, e^{i\omega t}\, s(t)\, dt \ . \qquad (1.3)$$

Thus, to obtain $F(\omega, b)$, one multiplies the signal by an appropriate window h (such as a Gaussian) centered at time b and then takes the Fourier transform. In mathematical terms, equation (1.3) is an expansion of the signal in terms of a family of functions $h(t-b)e^{i\omega t}$, which are generated from a single function h(t) through translations b in time and translations ω in frequency. In contrast, the wavelet transform (1.1) is an expansion in functions $g(\frac{t-b}{a})$ generated by translations b in time and dilations a in time.[3] Thus, the continuous wavelet transform resembles a (continuous) bank of short-time Fourier transforms with a different window for each frequency. The significance of this is that, while the basis functions in (1.3) have the same time and frequency resolution (that of h(t) and $\hat{h}(\omega)$) at all points of the transform plane, those of (1.1) have time resolution (that of $g(t/a)$) which decreases with a and frequency resolution (that of $\hat{g}(a\,\omega)$) which increases with a. This property can be a great advantage in signal processing since high-frequency signal

[2] The energy density is given by $| W |^2 \frac{da\, db}{a^2}$, an expression which is intimately linked to representations of the affine group (cf., [3], [15] – [16]). Since a is proportional to bandwidth, $| W |^2$ is power/Hz. See section 6 for a discussion of the discrete case.

[3] Alternatively, dilations in time may be considered contractions in frequency since the Fourier transform of $g(t/a)$ is $a\hat{g}(a\omega)$.)

characteristics are generally highly localized in time whereas slowly varying signals require good low-frequency resolution.

As originally proposed by Morlet et al. [17], g was a modulated Gaussian

$$g(t) = e^{i\nu_0 t} e^{-t^2/2} , \tag{1.4}$$

and this function is still the prototypical analyzing wavelet for signal-processing applications. [1] The window function $g(t/a)$ has Fourier transform $\hat{g}(a\omega) = a e^{-(\omega - \frac{\nu_0}{a})^2 a^2/2}$, which has analysis frequency ν_0/a. We emphasize that ν_0 is simply a parameter which determines the analyzing wavelet; its role should not be confused with that of a even though the scale axis is often expressed in terms of frequency under the transformation $a \rightarrow \nu_0/a$. Observe that, (1.4) only satisfies the admissibility condition (1.2) approximately (cf., [16], [18]). As expected its bandwidth is proportional to $1/a$, thus giving rise to a constant relative bandwidth; i.e., $BW/(\nu_0/a) = \text{constant}$. This feature is also reflected in the narrowing of the time window at higher frequencies; i.e., at smaller a. In general, one's choice of the function g is dictated by its time and frequency localization properties (see [3], [18]).

We shall be exclusively concerned with discrete values for a and b. In particular, we assume that a has the form $a = 2^i$ where i is termed the octave of the transform. The integral (1.1) then yields a *wavelet series*

$$W(2^i, n) \triangleq \frac{1}{\sqrt{2^i}} \int \bar{g}(\frac{t-n}{2^i})\, s(t)\, dt \quad . \tag{1.5a}$$

We remark that finite energy for the wavelet transform is not at all equivalent to finite energy for the wavelet series. It depends on the sampling grid as well as the function g(t). [3] Thus, the admissibility condition (1.2) is not necessarily appropriate in the discrete case and shall be replaced with conditions on the relevant filters in section 6. In addition, we shall often take b to be a multiple of a[4]

$$W(2^i, 2^i n) \triangleq \frac{1}{\sqrt{2^i}} \int \bar{g}(\frac{t}{2^i} - n)\, s(t)\, dt \quad . \tag{1.5b}$$

A logical step in applying the theory to discrete signals is to discretize the integral in (1.5)

[4] Physically, this reflects a need for less frequent sampling of the transform output at lower frequencies (i.e., larger scales a). Mathematically, $b = 2^i n$ has its roots in the orthonormal wavelets where it suffices for invertibility of the transform. [6] The general case, however, is much more complex. [3] Too sparse a sampling leads to incompleteness; oversampling results in a redundant set of functions.

$$w(2^i, 2^i n) \triangleq \frac{1}{\sqrt{2^i}} \sum_k \tilde{g}(\frac{k}{2^i} - n)\, s(k) \quad . \tag{1.6}$$

The sample rate has been set equal to one. As indicated by $2^i n$ on the left hand side, (1.6), as well as (1.5b), are decimated wavelet transforms. Octave i is only output every 2^i samples. In this form the resulting algorithms will not be translation invariant. [7] This is easily seen by substituting $s(k-r)$ for $s(k)$ which produces $w(2^i, 2^i(n-r/2^i))$, an integer translation of $w(2^i, 2^i n)$ only if r is a multiple of 2^i. However, the invariance, which is lost by decimation, is easily restored by separately filtering the even and odd sequences (cf., section 3) or by using an equivalent algorithm, also described in section 3. Our major reason for starting with (1.5b) rather than (1.5a) is historical. It delineates the relationship of the DWT to the QMF filter banks and (orthonormal) wavelet structures already found in the literature. [6] – [9] It also simplifies our derivation of the *'a trous* algorithm and readily lends itself to physical interpretation (cf., section 6). Note that the original Mallat algorithm [6] was decimated; *'a trous* [4] was not.

Proceeding from (1.6), we shall be able to to arrive at the DWT pictured in figure 1.1,

$$\begin{aligned}[s^i]_n &= \sum_j f_{2n-j}\, [s^{i-1}]_j \\ [w^i]_n &= \sum_j g_{n-j}\, [s^{i-1}]_j \quad .\end{aligned} \tag{1.7}$$

where $[w^i]_n$ corresponds to $w(2^i, 2^i n)$ of *equation (1.6) and* s^0 *is the original signal* s. The mysterious appearance of the filter f in (1.7) will be unraveled in the derivation of the *'a trous* algorithm in section 2. Finally, we shall come full circle in section 4 where, under quite general conditions, we show the existence of a function g(t) with $g(n) = g_n^\dagger \triangleq \bar{g}_{-n}$ and such that the DWT acting on the sampled signal is exactly the sampled output of the continuous wavelet transform (i.e., of the wavelet series).[5] In other words, the DWT with filter g defined by $g_n^\dagger \triangleq g_A(n)$, which was originally conceived as an approximation of the (continuous) WT for an arbitrary analyzing wavelet $g_A(t)$, is exact for another wavelet function $g_B(t)$ where $g_B(n) = g_n^\dagger$ for all n. Of course, if there is sufficient regularity, $g_A(t)$ and $g_B(t)$ will be close since they coincide on the integers up to the length of g.

Before embarking on this voyage, we summarize, and hopefully clarify, the plethora of transforms with a brief analogy to the Fourier transform, Fourier series, the discretized z-transform, and the discrete Fourier transform (DFT). The Fourier transform of a continuous signal s(t),

$$S(\omega) \triangleq \int_{-\infty}^{\infty} e^{-i\omega t}\, s(t)\, dt \quad , \tag{1.8}$$

[5] The adjoint filter $g_k^\dagger = \bar{g}_{-k}$ is used to simplify future notation. It corresponds to the integrand $\bar{g}(-t) * s$ found in equations (1.1), (1.5), and (1.6).

is a function of the continuous variable ω. Restricting it to a discrete (one-dimensional) grid results in the coefficicnts of a Fourier series

$$S(2\pi m) = \int_{-\infty}^{\infty} e^{-2\pi imt}\, s(t)\, dt, \tag{1.9}$$

which in turn may be computed approximately by

$$s_z(2\pi m\Delta t) = \sum_k e^{2\pi imk\Delta t}\, s(k\Delta t)\Delta t, \tag{1.10}$$

the z-transform of $s_n \triangleq s(n\Delta t)$ output at discrete points $e^{2\pi im\Delta t}$. If s(t) is band-limited and sampled at an appropriate rate, $\Delta t = 1/N$, then the above may be computed exactly using the DFT

$$\hat{s}_m \triangleq \frac{1}{N} \sum_1^N e^{\frac{2\pi imk}{N}}\, s_k \quad . \tag{1.11}$$

These correspond precisely to $W(a, b)$, $W(2^i, n)$, $w(2^i, n)$, and undecimated w_n^i. With wavelets, however, we have the additional difficulty of dealing with a whole class of functions g(t) rather than simply $e^{i\omega t}$. Also complicating things are its two-dimensional structure and the decimated versions, which, due to their $2^i n$ dependency on i, play a distinguished role without analogy in the one-dimensional case.

2. TWO ALGORITHMS

NOTATION

Decimation, which appears as a down arrow in figure 1.1, plays a pivotal role in all DWT algorithms. However, it leads to operators which are not time invariant and present a potential source of confusion. It is thus worthwhile to first establish some formal notation.

Signals and filters in boldface will be treated as vectors, in which case $*$ indicates discrete convolution and yields a vector. The symbol † will be used for adjoint filter $(\mathbf{f}^{\dagger})_k = \bar{f}_{-k}$. Note that this is the complex conjugate reversal and does not imply a conversion of a column vector to a row vector. The above mentioned decimation operator is represented by a matrix

$$\begin{aligned}\Delta_{kj} &\triangleq \delta(2k - j) \\ &= \delta_{2k,j}\end{aligned} \tag{2.1}$$

where δ_{ij} is the Kroneker delta and $\delta(k) \triangleq \delta_{k0}$. Also of significance is the dilation operator

$$\begin{aligned}D_{kj} &\triangleq \delta(k - 2j) \\ &= \delta_{k,2j} \ ,\end{aligned} \tag{2.2}$$

which dilates a vector by inserting zeros. Observe that Δ and $\mathbf{D}$ are transposes of each other, and that although they are linear, they are not time invariant; i.e., they are not functions of $k - j$.

Convolution followed by decimation becomes $[\Delta(\mathbf{f} * \mathbf{s})]_k = \sum_j \Delta_{kj} [\mathbf{f} * \mathbf{s}]_j = [\mathbf{f} * \mathbf{s}]_{2k} = \sum_m f_{2k-m} s_m$. However, a particularly insidious pitfall remains, namely,

$$(\Delta \mathbf{f}) * s \neq \Delta(\mathbf{f} * s) \, . \tag{2.3}$$

This associativity problem may be avoided by replacing convolution by $\mathbf{f}$ with a matrix, $\mathbf{F}$, defined by

$$F_{ij} \triangleq f_{i-j} \ , \tag{2.4}$$

which shall occasionally be used in our proofs. A trivial calculation yields $\Delta \mathbf{F} s = \Delta(\mathbf{f} * s)$. The symbol † will also be used for the adjoint of matrices. This is consistent with the above notation where $(F^{\dagger})_{ij} \triangleq \bar{F}_{ji} = \bar{f}_{j-i} \triangleq f^{\dagger}_{i-j}$.

We define the Fourier transform of a function s(t) by

$$\hat{s}(\omega) \triangleq \int s(t)\, e^{-i\omega t}\, dt \quad , \tag{2.5}$$

and the z-transform (on the unit circle) of a discrete signal **s** by

$$s_z(\omega) \triangleq \sum_n s_n\, e^{-i\omega n} \quad . \tag{2.6}$$

In the subsequent interplay between continuous and discrete functions one must be careful to distinguish the usage of these two transforms. Ignoring their differences can easily lead to erroneous conclusions. In particular,

$$(\Delta \mathbf{f})_z(\omega) = \sum_n f_{2n}\, e^{-in\omega} = \sum_{n \text{ even}} f_n\, e^{-i2n\omega} \neq \mathbf{f}_z(2\omega) \quad . \tag{2.7}$$

Some comment concerning filter definitions is also appropriate. Usage in the literature is uniform only up to the adjoint. Also, the z-transform is sometimes defined with a positive exponential which leads to similar differences in the frequency domain. In keeping with signal-processing applications we have chosen (2.6) as above, consistent with the Fourier transform, and we shall define our filters so that adjoints do not appear in convolutions. This produces a minimum of adjoints and greatly simplifies the notation. Unfortunately, it also results in the definition $g_n^\dagger = g(n)$ and the introduction of $\mathbf{f}^\dagger$ as an interpolation filter whereas **g** and **f** would be more natural. Note, also, that our filters are the adjoints of the filters defined by I. Daubechies [8], although their z-transforms coincide since [8] defines the z-transform with a plus sign.

THE `A TROUS ALGORITHM

We take the discretized wavelet series (1.6) as our starting point. The difficulty in implementing (1.6) is that, even for g(t) of finite support, as i increases, $\bar{g}$ must be sampled at progressively more points, creating a large computational burden. The solution posed in [4] is to approximate the values at nonintegral points through interpolation via a finite filter $\mathbf{f}^\dagger$.[6] The resulting recursion is highly efficient and may be implemented with the filter bank structure of figure 1.1.

The interpolation is perhaps best introduced with an example. Let $\mathbf{f}^\dagger$ be the filter (0.5, 1.0, 0.5). Then,

$$\sum_k f_{n-2k}^\dagger\, g(k) = \begin{cases} g(\frac{n}{2}) & \text{n even} \\ \frac{1}{2}(\, g(\frac{n-1}{2}) + g(\frac{n+1}{2})\,) & \text{n odd} \end{cases} \tag{2.8}$$

approximates a sampling of g(t/2). With the help of the dilation operator **D**, this may

[6] Again, we use the adjoint in our definition to simplify subsequent notation.

be formalized as a general procedure for dyadic interpolation. The steps are illustrated in figure 2.1. Let **g** be a filter defined by $g^{\dagger}_n \triangleq g(n)$. First, we spread $\mathbf{g}^{\dagger}$ to provide space in which to put the interpolated values. The resulting filter is $\mathbf{Dg}^{\dagger}$. Then, we apply a filter $\mathbf{f}^{\dagger}$ which leaves the even points fixed and interpolates to get the odd points. This condition, that **f** be the identity on even points, is sufficiently important to warrant a separate definition:

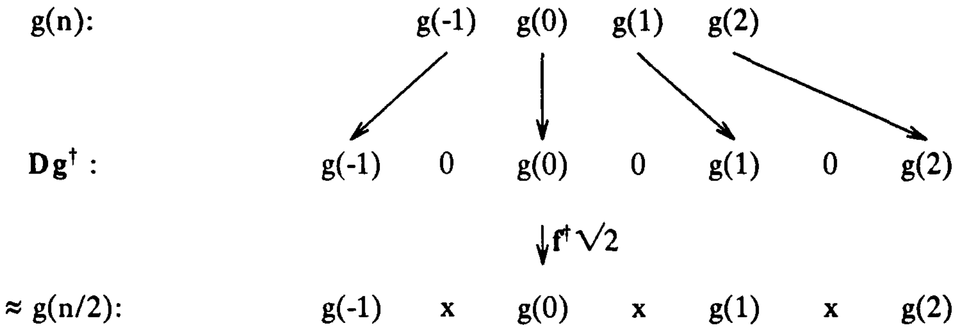

Figure 2.1 Diagram illustrating the dialation and interpolation of a function g(t) : $g(n/2) \approx \sqrt{2}\mathbf{f}^{\dagger} * (\mathbf{Dg}^{\dagger})$.

Definition 2.1

The lowpass filter **f** is said to be an *`a trous filter* if it satisfies

$$f_{2k} = \delta(k)/\sqrt{2} \ . \tag{2.9}$$

The $\sqrt{2}$ is simply a convenient means of including the normalization factor of (1.6) in the filter. The result of the entire interpolation operation, as pictured in figure 2.1, is thus

$$\begin{aligned} [\, \mathbf{f}^{\dagger} * (\mathbf{Dg}^{\dagger})\,]_n &= [\, \mathbf{F}^{\dagger}\mathbf{Dg}^{\dagger}\,]_n \\ &= \sum_k f^{\dagger}_{n-2k}\, g(k) \\ &\approx \frac{1}{\sqrt{2}}\, g(n/2) \ . \end{aligned} \tag{2.10}$$

Noting that $g(\frac{k}{2} - n) = g(\frac{k-2n}{2})$ and inserting the approximation (2.10) into (1.6), we obtain

$$
\begin{aligned}
w(2, n) &\approx \sum_{k,m} f^{\dagger}_{k-2n-2m}\, \bar{g}^{\dagger}_{m}\, s_k \\
&= \sum_{k,m'} g_{n-m'}\, f_{2m'-k}\, s_k \\
&= [\mathbf{g} * (\Delta(\mathbf{f} * \mathbf{s}))]_n \qquad (2.11)
\end{aligned}
$$

which is simply w^i_n of equation (1.7) with $i = 1$. Continuing inductively by replacing s in (2.11) with s^{i-1}, we find that $w(2^i,2^i n) \approx w^i_n$ for all i, where w^i_n is given by (1.7), which can be rewritten

$$
\begin{aligned}
\mathbf{s}^{i+1} &= \Delta\,(\mathbf{f} * \mathbf{s}^i) \qquad (2.12a) \\
\mathbf{w}^i &= \mathbf{g} * \mathbf{s}^i \ . \qquad (2.12b)
\end{aligned}
$$

Except for decimation of the output (the undecimated version will be derived in section 3), this is the *'a trous* algorithm described in [4]. Thus, we see that the *'a trous* algorithm is simply a DWT for which the filter **f** (an interpolator) satisfies condition (2.9) and the filter **g** is obtained by sampling an a priori wavelet function g(t).

Remark 1

The definition (1.6) is not so transparent as it might seem. It is, of course, intended to reflect an approximation to (1.5). From this viewpoint one might well consider a change of variables $t \rightarrow t/2^i$ before discretizing (1.5). Such a procedure certainly alleviates the computational problem since it dilates s(t) (that is, samples s at 2^i, values which are known) rather than contracting g(t). However, unless the original function s(t) was highly oversampled (which begs the computational question), the approximation is poor. More precisely, to accurately approximate s(t), and therefore also (1.5), we must sample at least at the Nyquist rate r_{nyq} for s. Then the integral for octave i requires g(t) to be sampled at a rate $2^i r_{nyq}$.

Remark 2

The derivation above, as well as that in [4], of the *'a trous* algorithm make no statements regarding the accuracy of the approximation (2.11) or even of the discretization from (1.5) to (1.6). The former is iterated over i and, hence, to succeed must be numerically stable in some sense. This, in turn depends principally on choice of the filter **f**. A major step towards treating this question lies in the results of section 4, as was outlined at the end of the introduction. Since the algorithm is exact for some $\tilde{g}(t)$, the question reduces to (a) the quality of the approximation $g \approx \tilde{g}$ and (b) the effect of this approximation on the wavelet integral (1.1). Inasmuch as $g(n) = \tilde{g}(n)$ for the finite set of integers n used to obtain **g** from g(t), it is sufficient that these functions be smooth enough and decrease fast enough at infinity. Conditions on **f** which achieve regularity of the constructed wavelet $\tilde{g}$ are found in [8]. Quantifying these statements remains a subject for future study.

MULTIRESOLUTION ALGORITHM

Mallat's algorithm, illustrated in figure 2.2, has essentially the the same tree structure as equations (2.12) (cf., [6] – [9]), namely,

$$s^{i+1} = \Delta\,(\mathbf{h} * s^{i}) \tag{2.13a}$$

$$\mathbf{d}^{i+1} = \Delta\,(\mathbf{g} * s^{i}) \tag{2.13b}$$

In keeping with the literature, we have replaced the filter **f** with the filter **h**, which also serves to indicate that this class of filters is constrained, as detailed below. We remark that none of these filters are *ˋa trous* filters. The constraints on **h** and **g** which insure an orthonormal multiresolution analysis (cf., [6] – [9]) are

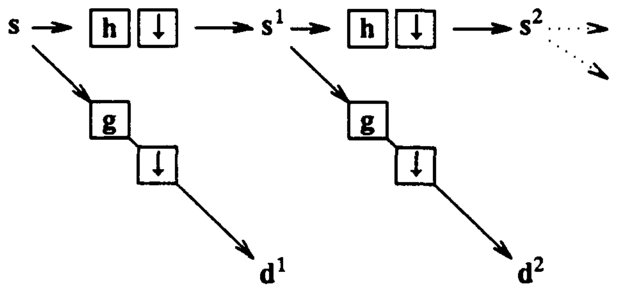

Figure 2.2 The Mallat multiresolution algorithm. The down-arrow indicates decimation.

$$\sum_j \bar{h}_{2j-n}\, h_{2j-m} + \bar{g}_{2j-n}\, g_{2j-m} = \delta_{nm} \tag{2.14a}$$

$$\sum_j h_{2n-j}\,\bar{g}_{2m-j} = 0 \tag{2.14b}$$

$$\sum_n g_n = 0 \tag{2.14c}$$

$$\sum_n h_n = \sqrt{2}\ \ . \tag{2.14d}$$

Recalling that $H_{ij} \triangleq h_{i-j}$ and that $\Delta^{\dagger} = \mathbf{D}$, we may rewrite (2.14a) and (2.14b) as

$$(\mathbf{H}^{\dagger}\mathbf{D})\,(\Delta\mathbf{H}) + (\mathbf{G}^{\dagger}\mathbf{D})\,(\Delta\mathbf{G}) = \mathbf{I} \tag{2.15}$$

$$(\Delta\mathbf{H})\,(\mathbf{G}^{\dagger}\mathbf{D}) = 0\ \ . \tag{2.16}$$

Furthermore, (2.15) and (2.16) imply (e.g., multiply (2.15) on the left by $\Delta H^\dagger$)

$$\begin{aligned}(\Delta H)\,(H^\dagger D) &= I \\ (\Delta G)\,(G^\dagger D) &= I \end{aligned} \tag{2.17}$$

Thus, $H^\dagger D$ and $G^\dagger D$ are injections and (2.13) is an orthogonal decomposition of the discrete signal s^i. That is, $s^{i-1} = H^\dagger D s^i + G^\dagger D w^i$ with the scalar product $(H^\dagger D s^i) \cdot (G^\dagger D w^i) = 0$. In fact, equation (2.15) is a paradigm for inverting the transform. These concepts are illustrated in figure 2.3.

$$\begin{tikzcd} s^{i-1} \arrow[r, "\Delta H"] \arrow[d, "\Delta G"'] & s^i & & s^{i-1} & \oplus \arrow[l] & s^i \arrow[l, "H^\dagger D"'] \\ d^i & & & & d^i \arrow[u, "G^\dagger D"] & \end{tikzcd}$$

Figure 2.3 Illustration of a single stage and its inverse in the multiresolution algorithm for orthonormal wavelets.

Furthermore, from (2.14) it follows that (2.13) represents a wavelet decomposition (see [6], [8], [9]) as follows: There exists a scaling function $\phi(t)$ whose Fourier transform is given by

$$\hat{\phi}(\omega) \triangleq \prod_{j=1}^{\infty} \left[\frac{1}{\sqrt{2}}\, \overline{h}_z(\frac{\omega}{2^j}) \right] . \tag{2.18}$$

Expression (2.18) forms the basis of a recursion,

$$\hat{\phi}(\omega) = \frac{1}{\sqrt{2}} \overline{h}_z(\frac{\omega}{2})\, \hat{\phi}(\frac{\omega}{2}) , \tag{2.19}$$

which in the time domain takes the form

$$\phi(t) = \sum_k \overline{h}_{-k} \sqrt{2}\, \phi(2t-k) . \tag{2.20}$$

With some additional computation (cf., section 4), one may demonstrate that the translates and dilates of ϕ,

$$\phi_n^i(t) \triangleq \frac{1}{\sqrt{2^i}}\,\phi(\frac{t}{2^i} - n) \quad , \tag{2.21}$$

have the property

$$\phi_n^{i+1}(t) = \sum_k [\Delta H]_{nk}\, \phi_k^i(t) \quad . \tag{2.22}$$

Note that the above definitions differ in the sign of i from those of [8].

Finally, define

$$\psi(t) \triangleq \; = \sum_k \sqrt{2}\, \bar{g}_{-k}\, \phi(2t - k) \quad . \tag{2.23}$$

Then, using (2.14) and the above properties of ϕ, one can show that the family of wavelets,[7]

$$\psi_n^i(t) \triangleq \frac{1}{\sqrt{2^i}}\, \psi(\frac{t}{2^i} - n) \tag{2.24}$$

are orthonormal ($\int \psi_n^i(t)\, \bar{\psi}_k^j(t)\, dt = \delta_{ij}\delta_{nk}$), and that the $\mathbf{d}^i$ are the coefficients of the expansion of s(t) in terms of the ψ_n^i.

More precisely, the integer translates $\phi(t - k)$ of the scaling function form a basis for $L^2(R)$, and

$$d_n^i = \frac{1}{\sqrt{2^i}} \int s(t)\, \bar{\psi}(\frac{t}{2^i} - n) = \int s(t)\, \bar{\psi}_n^i(t) \tag{2.25}$$

provided

$$s_k^0 = \int s(t)\, \bar{\phi}(t-k) \quad . \tag{2.26}$$

Then, if s(t) is in $L^2(R)$, the expansion of s(t) is

$$s(t) = \sum_{i\,n} d_n^i\, \psi_n^i(t) \quad . \tag{2.27}$$

This follows from (2.25) and orthonormality since completeness of the $\phi(t - k)$ implies completeness of the $\psi_n^i(t)$.

[7] In contrast, the $g_n^i(t) \triangleq \frac{1}{\sqrt{2^i}}\, g(\frac{t}{2^i} - n)$ of (1.5) are not generally orthogonal. It is the filter constraints (2.14a) and (2.14b) that insure orthonormality. Dropping these two constraints in section 4, we develop a structure identical to (2.21) – (2.24); however, the constructed $g_n^i(t)$ need not be orthogonal.

Until recently, the only known wavelets with compact support (i.e., zero outside a finite interval) were the Haar functions, generated by $\psi(t) = 1$ for $0 \leq t < 1/2$; -1 for $-1/2 \leq t < 0$; and 0 elsewhere. Ingrid Daubechies (cf., [8], [19]) has uncovered an entire family of finite length filters satisfying (2.14), demonstrating that the corresponding wavelets defined by (2.18), (2.23) and (2.24) are orthonormal as a consequence of (2.15) - (2.16) and have compact support since they were generated by finite length filters (cf., section 4, equation (4.10)). The first two of these filters are

$$\mathbf{h}^{\dagger} = \frac{1}{\sqrt{2}}\,(1,\,1)\,, \tag{2.28a}$$

and

$$\mathbf{h}^{\dagger} = \frac{1}{4\sqrt{2}}\,(1+\sqrt{3},\,3+\sqrt{3},\,3-\sqrt{3},\,1-\sqrt{3}) \tag{2.28b}$$

where the first component of **h** is on the left. The wavelets corresponding to (2.28a) are exactly the Harr function mentioned above.

Some additional remarks relating the two algorithms are in order. The conditions (2.14c) and (2.14d) effectively make **g** a bandpass filter and **h** a lowpass filter (e.g., an interpolation filter) with the sum on g_k analogous to the condition $\int g(t)dt = 0$. Also, $\mathbf{d}^{i+1}$ and $\mathbf{w}^{i}$ correspond; the additional decimation appearing in (2.13b) would not appear in a translation invariant version of Mallat's algorithm (cf., section 3). On the other hand, although the discrete filters **g** play algorithmically identical roles, the $g_n^i(t)$ of the *'a trous* algorithm are not the wavelet vectors of a functional expansion. Rather the $g_n^i(t)$ and $\psi_n^i(t)$ are the duals of a set of vectors for which coefficients of the signal expansion are w_{2n}^i and d_n^i respectively (cf., [3], [15]). That is, they are the coefficients of an expansion of the form $s(t) = \sum$ $<s,\, \psi_n^i>\, \tilde{\psi}_n^i(t)$ where $< >$ indicates the L^2 inner product, and $\tilde{\psi}_n^i$ is the dual basis or^{in} frame (see [3]) of ψ_n^i. In Mallat's algorithm, since the $\psi_n^i(t)$ are orthonormal, the basis and its dual coincide. Thus, in many senses, the discrete filters **g** are more fundamental than the wavelets themselves. Even in the orthonormal case, it is usually the coefficients which are of major interest; the actual wavelets $\psi_n^i(t)$ or the duals of the $g_n^i(t)$ are rarely computed.

Finally, in anticipation of section 5, let us form the squares of the two previous examples

$$\mathbf{h}^{\dagger} * \mathbf{h} = \frac{1}{\sqrt{2}}(1,\,1) * \frac{1}{\sqrt{2}}(1,\,1) = (1/2,\,1,\,1/2) \tag{2.29a}$$

and

$$\mathbf{h}^{\dagger} * \mathbf{h} = (-\frac{1}{16},\,0,\,\frac{9}{16},\,1,\,\frac{9}{16},\,0,\,-\frac{1}{16}) \tag{2.29b}$$

where $*$ indicates convolution, and **h** given by (2.28a) and (2.28b) respectively. Note that the filters $\mathbf{f} = \mathbf{h}^{\dagger} * \mathbf{h}\,/\sqrt{2}$ are *'a trous* filters and that the interpolation equation (2.10) holds exactly if g(t) is a polynomial in t of degree one or three respectively.

3. WAVELET TRANSFORMS WITHOUT DECIMATION

As has often been pointed out [7], the recursions (2.12) and (2.13) are not, in general, translation invariant. In contrast, the original undecimated *'a trous* algorithm (cf., [4], [5]), which is pictured in figure 3.1 and consists entirely of convolutions, is patently translation invariant. In this subsection we shall use that property to provide a formal definition for an undecimated discrete wavelet transform $\tilde{\mathbf{w}}$, then demonstrate that $w^i_n = \tilde{w}^i_{2^i n}$, and also show that $\tilde{\mathbf{w}}$ is computed by the algorithm of figure 3.1.

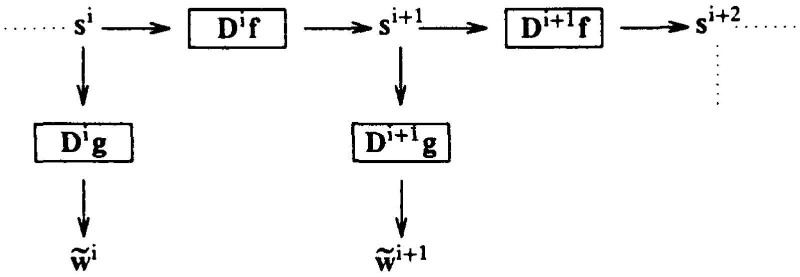

Figure 3.1 The (undecimated) discrete wavelet transform. The filters $\mathbf{D}^i\mathbf{f}$ are obtained from $\mathbf{f}$ by inserting 2^i-1 zeros between each pair of filter coefficients. The operation of filtering is understood to mean convolution.

Let T_m be the operation of translation by m; i.e.,

$$(T_m \mathbf{s})_n \triangleq s_{n-m} \qquad (3.1)$$

In order to include the dependency of $\mathbf{w}^i$ on $\mathbf{s}^0$, we add it as an argument, writing $\mathbf{w}^i(\mathbf{s}^0)$. Equations (2.12a) and (2.12b) become

$$\mathbf{w}^i(\mathbf{s}^0) = \mathbf{G}(\Delta\mathbf{F})^i\,\mathbf{s}^0 \ . \qquad (3.2)$$

(Recall that $\mathbf{G}_{ij} \triangleq g_{i-j}$.) Finally, we shall have need of the following important identities, which are proved in appendix A: For any $\mathbf{F}$ of the form $F_{ij} = f_{i-j}$,

Lemma 3.1

$$(\Delta\mathbf{F})^i_{mk} = (\Delta\mathbf{F})^i_{0,\,k-2^i m} \ , \qquad (3.3)$$

and

Lemma 3.2

$$\sum_k (\Delta F)^i_{nk}\, e^{ik\omega} = e^{i2^i n\omega} \prod_{j=0}^{i-1} f_z(2^j\omega) \quad . \tag{3.4}$$

As expected, w^i is not translation invariant,

$$\begin{aligned}[w^i(T_m s^0)]_n &= \sum_k [G(\Delta F)^i]_{nk}\, s^0_{k-m} \\ &= \sum_k [G(\Delta F)^i]_{n,k+m}\, s^0_k \\ &\neq \sum_k [G(\Delta F)^i]_{n-m,k}\, s^0_k \quad .\end{aligned} \tag{3.5}$$

For example, $[\Delta F]_{ij} = f_{2i-j} \neq [\Delta F]_{0,i-j}$. However, if we replace m by 2^im in (3.5) and use Lemma 3.1, the last step becomes an equality so that

$$[w^i(T_{2^i m} s^0)]_n = [w^i(s^0)]_{n-m} \quad . \tag{3.6}$$

Thus, translating s^0 by 2^im translates octave i by m.

Note that the zeroth element of a series is invariant under decimation so that w^i_n and $\tilde{w}^i_n$ should coincide at $n = 0$. Using this fact, we obtain the nth output of the undecimated discrete wavelet transform by translating the signal back by n samples and taking the decimated transform at time zero. More precisely,

Definition 3.1

Define the undecimated discrete wavelet transform $\tilde{w}$ in terms of the decimated transform w by

$$\tilde{w}^i_n \triangleq [\tilde{w}^i(s^0)]_n \triangleq [w^i(T_{-n} s^0)]_0 \quad , \tag{3.7}$$

We see that the desired invariance is achieved,

$$\begin{aligned}[\tilde{w}^i(T_m s^0)]_n &= [w^i(T_{-n} T_m s^0)]_0 \\ &= [w^i(T_{m-n} s^0)]_0 \\ &= [\tilde{w}^i(s^0)]_{n-m} \quad .\end{aligned} \tag{3.8}$$

It is also clear from (3.6) and (3.7) that sampling $\tilde{w}^i_n$ every 2^i points produces exactly w^i_n, that is,

$$w^i_n = \tilde{w}^i_{2^i n} \quad . \tag{3.9}$$

Next, we show that $\tilde{\mathbf{w}}$ may be computed by the filter sequence pictured in figure 3.1. The proof is obtained by taking z-transforms. From (3.2) and (3.7)

$$\begin{aligned}\sum_n \tilde{w}^i_n e^{-in\omega} &= \sum_{mn} [G(\Delta F)^i]_{0m}\, s_{m+n}\, e^{-in\omega} \\ &= \sum_m [G(\Delta F)^i]_{0m}\, e^{im\omega}\, s_z(\omega) \ . \qquad (3.10)\end{aligned}$$

Applying Lemma 3.2 to (3.10), we have

$$\begin{aligned}\tilde{\mathbf{w}}^i_z(\omega) &\triangleq \sum_n \tilde{w}^i_n e^{-in\omega} \\ &= \mathbf{g}_z(2^i\omega) \prod_{j=0}^{i-1} \mathbf{f}_z(2^j\omega)\, s_z(\omega) \qquad (3.11)\end{aligned}$$

where i = 0 is understood to mean there are no factors of $\mathbf{f}_z$. As described in the next paragraph, this is exactly the z-transform of the algorithm pictured in figure 3.1.

It is easy to see that $D^i\mathbf{f}$ is $\mathbf{f}$ with 2^i-1 zeros inserted between every pair of filter coefficients and that its z-transform is $\mathbf{f}_z(2^i\omega)$. That is,

$$[D^i\mathbf{f}]_n = \begin{cases} f_{n/2^i} & n = 2^i m \\ 0 & \text{other} \end{cases} \qquad (3.12)$$

and

$$(D^i\mathbf{f})_z = \mathbf{f}_z(2^i\omega) \ . \qquad (3.13)$$

Equation (3.11) is then equivalent to

$$\mathbf{s}^{i+1} = (D^i\mathbf{f}) * \mathbf{s}^i \qquad (3.14a)$$

$$\tilde{\mathbf{w}}^i = (D^i\mathbf{g}) * \mathbf{s}^i \qquad (3.14b)$$

where $\mathbf{s}^0 \triangleq \mathbf{s}$. This is essentially the original (undecimated) ‘*a trous* algorithm found in [4] and [5]. However, we would like to emphasize that, since the development in this subsection has not made use of any filter constraints, the general equivalence of the decimated output of the algorithm pictured in figure 3.1 (equations (3.14)) and that of figure 1.1 (equations (2.12)) follows for arbitrary filters.

An alternative implementation

A second possibility for the implementation of $\tilde{\mathbf{w}}$ is to use the algorithm in figure 1.1 and proceed directly from equation (3.7). That is, to output $\tilde{w}^i_n$, one simply translates the signal by n and then computes w^i_0. As an algorithm, this takes the form (a) compute w^i_0; (b) translate the signal by one sample; (c) go to step (a). Moreover, in view of (3.6), one need not reperform the entire recursion (i.e., equations (2.12)) for every time point n in order to obtain $\tilde{w}^i$. Rather, at each octave, the decimation is replaced by a split into even and odd sequences, each of which is a starting point for the next octave (cf., figure 3.2). A few examples suffice to convince one that if, at octave i, n mod $2^i = 0$, then one takes the upper branch; if n mod $2^i = 1$, then one takes the lower branch. A rigorous derivation follows from the formula (cf., [20]).

$$\Delta T^m s = \begin{cases} T^{\frac{m}{2}} \ s_{even} & m \text{ even} \\ T^{\frac{m-1}{2}} \ s_{odd} & m \text{ odd} \end{cases} \tag{3.15}$$

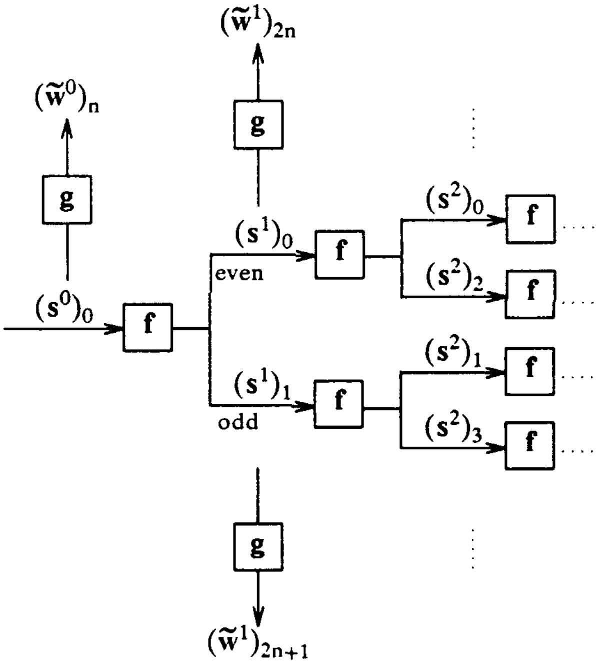

Figure 3.2 Diagram of an implementation of the undecimated DWT.

We remark that figures 5 and 6 are computationally equivalent provided that the algorithm in figure 3.1 is implemented efficiently. The code must be written so as to omit multiplication by the zero elements of filters $\mathbf{D}^i\mathbf{f}$. (They are mostly zeros for i > 2.) Depending on the number of octaves, the computational burden still remains much greater than that of the decimated algorithm (i.e., figure 1.1); however, there is considerable parallelism which may be sufficiently exploited on suitable hardware to produce a realtime implementation. [5]

4. THE DWT AS AN EXACT WAVELET TRANSFORM

Regardless of the filters used, one can, of course, perform the recursions (2.12) or (2.13) on the sampled signal **s**. Moreover, provided that **f** (respectively, **h**) is lowpass and **g** bandpass, the procedure may be interpreted physically as a bank of proportional bandwidth filters (cf., [21] – [24], also section 6). In the present section, we examine the mathematical significance of relaxing the filter constraints (2.9) and (2.14). Our goal will be to relate the more general filter bank to the continuous wavelet transform, thus, in a sense, justifying the term DWT. In this endeavor, the major questions we shall address are: for what functions g(t) is the recursion (2.12) an exact implementation of (1.6) and for which g(t) and s(t) do (1.5) and (1.6) coincide? The general answer is that we are able to construct such a **g** provided the discretized signal lies in the appropriate subspace (cf., equation (2.26)). A somewhat surprising result is that it is necessary and sufficient for **f** to be *'a trous* for condition (2.26) to be dropped. Our approach shall be to mimic the construction of orthonormal wavelets outlined in section 2.

CONSTRUCTION OF THE SCALING FUNCTION ϕ

We begin with the stipulation of the existence of a scaling function $\phi(t)$ with Fourier transform

$$\hat{\phi}(\omega) \triangleq \prod_{j=1}^{\infty} \left[\frac{1}{\sqrt{2}} \overline{\mathbf{f}}_z(\frac{\omega}{2^j}) \right] \tag{4.1}$$

where $\overline{\mathbf{f}}_z(\omega) = (\mathbf{f}^{\dagger})_z(\omega)$ is the z-transform of $\mathbf{f}^{\dagger}$. To emphasize the nonorthogonality of the corresponding wavelets, we retain the symbol **f** rather than **h**. Note that this function ϕ need not have (and in general does not have) all of the properties of the orthonormal ϕ outlined at the end of section 2.

For (4.1) to converge to a nonzero function, the factors must approach one. Thus, $\mathbf{f}_z(0) = 1$, which implies

$$\sum_k f_k = \sqrt{2} \quad . \tag{4.2}$$

Even though ϕ could be normalized differently by the inclusion of a factor in (4.1), the filter **f** must necessarily obey the lowpass condition (4.2); i.e., (2.14d). Note also that, under the chosen normalization, $\int \phi(t)dt = \hat{\phi}(0) = 1$. However, without some additional conditions, the relationship of $\phi(t)$ to $\hat{\phi}(\omega)$ remains somewhat tenuous. Even under pointwise convergence, the limit may be a, highly discontinuous, fractal function. [9] Suitable regularity conditions for the inverse Fourier transform of a product of the form (4.1) to converge to a reasonably behaved (e.g, $L^1(R)$, $L^2(R)$, and/or continuous) function may be found in [8] and [25]. The results are summarized in appendix B.

A time domain representation of $\phi(t)$ in terms of **f** can be derived from (4.1). Let χ be the indicator function of $[-1/2, 1/2)$

$$\chi(t) \triangleq \begin{cases} 1 & \text{for } t \in [-1/2,\, 1/2) \\ 0 & \text{other} \end{cases} \quad . \tag{4.3}$$

From (4.1) and Lemma 3.2, it is not difficult to see that

$$\phi(t) = \lim_{j\to\infty} \sum_k ((\Delta F)^j)_{0k} \sqrt{2^j}\,\chi(2^j t - k) \ . \tag{4.4}$$

In fact, the Fourier transform of (4.4) is[8]

$$\lim_{i\to\infty} \sum_k (\Delta F)^i_{0k}\, e^{-k\frac{\omega}{2^i}} \frac{1}{\sqrt{2^i}} \hat{\chi}(\frac{\omega}{2^i}) = \lim_{i\to\infty} \prod_{j=1}^{i} \left(\frac{1}{\sqrt{2}} \bar{f}_z(\frac{\omega}{2^j})\right) \frac{\sin(\omega/2^{i+1})}{\omega/2^{i+1}} \tag{4.5}$$

which is just (4.1). Note that the existence of the function (4.4) could have been taken as the starting point for our analysis, since our proofs will not make use of (4.1).

We continue our parallel with the development of Mallat's algorithm in section 2. Define $\phi^i_n(t)$ by

Definition 4.1

$$\phi^i_n(t) \triangleq \frac{1}{\sqrt{2^i}} \phi(\frac{t}{2^i} - n) \ . \tag{4.6}$$

Then, substitution of $2^i t - n$ for t in (4.4) and the use of Lemma 3.1 yield

$$\begin{aligned} \sqrt{2^i}\,\phi(2^i t - n) &= \lim_{j\to\infty} \sum_k ((\Delta F)^j)_{0k} \sqrt{2^{j+i}}\phi(2^{j+i}t - 2^j n - k) \\ &= \lim_{j\to\infty} \sum_k ((\Delta F)^{j-i})_{0,k-2^{j-i}n} \sqrt{2^j}\phi(2^j t - k) \\ &= \lim_{j\to\infty} \sum_k ((\Delta F)^{j-i})_{nk} \sqrt{2^j}\chi(2^j t - k) \ . \end{aligned} \tag{4.7}$$

On replacing i by –i, this becomes

$$\phi^i_n(t) = \lim_{j\to\infty} \sum_k ((\Delta F)^{j+i})_{nk} \sqrt{2^j}\chi(2^j t - k) \ . \tag{4.8}$$

[8] This equality is immediate from Lemma 3.2 and $\hat{\chi}(\omega) = (2\sin\frac{\omega}{2})/\omega$.

An immediate consequence of (4.8) is

$$\phi_n^{i+1} = \sum_k [\Delta \mathbf{F}]_{nk}\, \phi_k^i \tag{4.9}$$

paralleling (2.22). Thus, we see that, despite their lack of orthogonality, the $\phi_n^i(t)$ have retained most of their structure. Furthermore, if **f** is a finite filter, then $\phi(t)$ has finite support. [8] More, precisely, suppose the coefficients of **f** are zero outside $[-N_-, N_+]$. Let $\eta^j(t) \triangleq ((\Delta F)^j)_{0k} \sqrt{2^j}\, \chi(2^j t - k)$. Then, $\eta^j(t)$ converges to $\phi(t)$, and, as in (4.7), we have

$$\eta^{j+1}(t) = \sum_k f_k^\dagger\, \eta^j(2t-k) \quad . \tag{4.10}$$

We find, for example, that η^j is zero for $t < t_j$ where $t_j = (t_{j-1} - N_-)/2$. With $t_0 = -1/2$, and with a similar calculation for the right half interval, it follows that $\phi(t) = \lim\limits_{j\to\infty} \eta^j(t)$ is zero outside $[-N_-, N_+]$.

EXACTNESS

To avoid confusion and stress their differences, let us first recapitulate some definitions. Four different transforms $W(a, b)$, $W(2^i, 2^i n)$, $w(2^i, 2^i n)$, and w_n^i have been mentioned ((1.1), (1.5), (1.6) and (2.12)). We retain a terminology parallel to Fourier transforms, namely, wavelet transform (WT), wavelet series[9], discretized wavelet series, and discrete wavelet transform (DWT) respectively. The first two transforms involve integrals of a continuous signal; the latter two contain sums of sampled signals. The first three utilize a continuous wavelet function g(t), the last one employs the discrete filters **g** and **f**. For consistency, we shall continue our development using decimated transforms; however, the results hold without change for undecimated transforms. This follows immediately, since they coincide for $n = 0$, and the undecimated transforms may be obtained at time $n = n_0$, by translating the signal by n_0 samples and taking the transform at $n = 0$. (See section 3 where definition 3.1 remains valid for $W(2^i, n)$ and $w(2^i, n)$).

Our starting point shall be a signal s(t) and discrete filters **f** and **g** with $\mathbf{w}^i$ defined by (2.12) or $\mathbf{d}^i$ by (2.13), i.e.,

$$\begin{aligned} \mathbf{s}^{i+1} &= \Delta\, \mathbf{F}\, \mathbf{s}^i \\ \mathbf{w}^i &= \mathbf{G}\, \mathbf{s}^i \\ \mathbf{d}^{i+1} &\triangleq \Delta\, \mathbf{w}^i \quad . \end{aligned} \tag{4.11}$$

Recall that the matrices F_{ij} and G_{ij} are given by f_{i-j} and g_{i-j} respectively. Of course, we must also specify an initialization of the recursion (4.11) for some i; for example,

[9] At times, we shall prefer the term sampled WT rather than wavelet series in order to emphasize its role as a restriction of the continuous transform.

for the zeroth octave s^0. The obvious choice is

$$s_n^0 \triangleq s(n) \quad ; \tag{4.12a}$$

however, we shall also consider

$$s_n^0 \triangleq \sum_k \phi(k-n)\, s(k) \tag{4.12b}$$

which relates to the discretized wavelet series $w(2^i, 2^i n)$, and

$$s_n^0 \triangleq \int \phi(t-n)\, s(t)\, dt \tag{4.12c}$$

which corresponds to the sampled WT (wavelet series). For a given **g**, we shall construct a continuous function g(t) such that the DWT of equation (4.11) is an exact implementation of the discretized wavelet series under (4.12b) and of the wavelet transform under (4.12c).

Define g(t) by

$$g(t) \triangleq \sum_k \phi(t+k)\, \bar{g}_k = \sum_k \phi(t-k)\, g_k^{\dagger} \quad , \tag{4.13}$$

and $g_n^i(t)$ by

$$g_n^i(t) \triangleq \frac{1}{\sqrt{2^i}}\, g(\frac{t}{2^i} - n) \quad . \tag{4.14}$$

Then, using (4.6), (4.9), (4.13), and (4.14), we have

$$\begin{aligned} g_n^i(t) &\triangleq \frac{1}{\sqrt{2^i}}\, g(\frac{t}{2^i} - n) \\ &= \sum_k g_k^{\dagger} \frac{1}{\sqrt{2^i}}\, \phi(\frac{t}{2^i} - n - k) \\ &= \sum_k \bar{g}_{n-k}\, \phi_k^i(t) \\ &= \sum_k [\bar{\mathbf{G}}(\Delta\mathbf{F})^i]_{nk}\, \phi_k^0(t) \quad . \end{aligned} \tag{4.15}$$

If we take (4.12b) as the definition of s^0, the discretized wavelet series (equation (1.6)) takes the form

$$\begin{aligned} w(2^i, 2^i n) &\triangleq \sum_{mk} \bar{g}_n^i(m)\, s(m) \\ &= [\sum_m \bar{\mathbf{G}}\, (\Delta\mathbf{F})^i]_{nk}\, \phi_k^0(m)\, s(m) \end{aligned}$$

$$= [G\,(\Delta F)^i\, s^0]_n \qquad (4.16)$$

which, by (4.11) is exactly w_n^i . Furthermore, under (4.12c) we have (again using (4.15))

$$\begin{aligned} W(2^i, 2^i n) &\triangleq \int \bar{g}_n^i(t)\, s(t)\, dt \\ &= \int \sum_k [G(\Delta F)^i]_{nk}\, \phi_k^0(t)\, s(t)\, dt \\ &= \sum_k [G\,(\Delta F)^i)_{nk} \int \phi_k^0(t)\, s(t)\, dt \\ &= [G\,(\Delta F)^i\, s^0]_n \quad , \qquad (4.17) \end{aligned}$$

again w_n^i .

Finally, let us investigate the significance of the ‘*a trous* condition; i.e., of the constraint (2.9), $f_{2k} = \delta_{k0}/\sqrt{2}$. We prove the following theorem:

Theorem 4.1

f is an ‘*a trous* filter $\Leftrightarrow$ $\phi(n) = \delta_{n0}$.

Proof

Letting $i = -1$, $n = 0$, and $t = n$ in equation (4.9) gives

$$\phi(n) \;=\; \sum_k \,(\Delta F)_{0k}\, \sqrt{2}\, \phi(2n-k) \quad . \qquad (4.18)$$

Then,

$$\phi(n) = \delta_{n0} \;\Rightarrow\; \sqrt{2}\, f_{2n} = \delta_{n0} \quad . \qquad (4.19)$$

Conversely, suppose that $f_{2n} = \delta_{n0}/\sqrt{2}$. Then, since $\chi(-k) = \delta_{k0}$, equation (4.8) with $i = 0$ and $t = 0$ implies

$$\begin{aligned} \phi(-n) &= \lim_{j\to\infty} (\Delta F)^j{}_{n0}\, \sqrt{2}^j \\ &= \lim_{j\to\infty} \sum_k (\Delta F)^{j-1}{}_{nk}\, (\Delta F)_{k0}\, \sqrt{2}\sqrt{2}^{j-1} \\ &= \lim_{j\to\infty} (\Delta F)^{j-1}{}_{n0}\, \sqrt{2}^{j-1} \\ \ldots &= \lim_{j\to\infty} (\Delta F)_{n0}\sqrt{2} \;=\; \delta_{n0} \quad . \qquad (4.20) \end{aligned}$$

The import of theorem 4.1 quickly follows. Equation (4.12b) implies (4.12a) for arbitrary signals if and only if $\phi(n) = \delta_{n0}$. Thus, *Corollary 4.1*

The algorithm (4.11) is an exact implementation of the DWT with s^0 set equal to the sampled signal if and only if **f** is an *'a trous* filter.

Furthermore, **g** is also related to g(t) by sampling:

Corollary 4.2

If **f** is an *'a trous* filter, then g(t) defined by (4.13) satisfies $g(n) = g_n^\dagger$.

It remains to investigate the role of $\mathbf{d}^i$. Define

$$\psi^g(t) \triangleq \sqrt{2}\, g(2t) = \sqrt{2} \sum_k \overline{g}_k\, \phi(2t+k) \tag{4.21}$$

which coincides with (2.23). Then, provided s^0 satisfies (4.12c), (4.11), and (4.17) imply

$$\begin{aligned} d_n^i &= (\Delta w^{i-1})_n \\ &= w_{2n}^{i-1} \\ &= \int \overline{g}_{2n}^{i-1}(t)\, s(t)\, dt \\ &= \int \frac{1}{\sqrt{2}} (\overline{\psi}^g)_n^{i-1}(\frac{t}{2})\, s(t)\, dt \\ &= \int (\overline{\psi}^g)_n^i\, (t)\, s(t)\, dt \quad . \end{aligned} \tag{4.22}$$

Hence, although the $(\psi^g)_n^i(t)$ are not orthogonal, the Mallat algorithm still computes the wavelet transform. Of course, under (4.12a) and *'a trous*, or for (4.12b), we have the counterpart of (4.16)

$$d_n^i = \sum_k (\overline{\psi}^g)_n^i(k)\, s(k) \quad . \tag{4.23}$$

It is interesting that, in a sense, the decimated wavelet transforms (1.5b) and (1.6) contain superfluous information. That is, they are underdecimated by a factor of two, and, thus, w^i properly belongs to octave i + 1.

SUMMARY

Let us summarize the results of this section. We are given discrete filters **f** and **g** such that (4.4) is well defined. Define g(t) by

$$g(t) \triangleq \sum_k \phi(t-k)\, g_k^\dagger \tag{4.24}$$

with corresponding transforms sampled WT (wavelet series)

$$W(2^i, 2^i n) \triangleq \int \bar{g}^i_n(t)\, s(t)\, dt \tag{4.25}$$

and discretized wavelet series

$$w(2^i, 2^i n) \triangleq \sum_k \bar{g}^i_n(k)\, s(k) \quad . \tag{4.26}$$

Let $\phi_n \cdot s$ stand for the scalar product $\sum_k \phi_n(k) s_k$, and $\phi_n(t) \cdot s(t)$ for the L^2 scalar product $\int \phi_n(t)\, s(t)$. Then:

$$\mathbf{f} \text{ is } \grave{a}\ \textit{trous} \quad \Rightarrow \quad g(n) = g_n \tag{4.27}$$

For s discrete:

$$s^0_n = \phi_n \cdot s \quad \Rightarrow \quad w(2^i, 2^i n) = w^i_n \tag{4.28a}$$

$$s^0 = s \text{ and } \mathbf{f} \text{ is } \grave{a}\ \textit{trous} \quad \Rightarrow \quad w(2^i, 2^i n) = w^i_n \quad . \tag{4.28b}$$

For s(t) continuous:

$$s^0_n = \phi_n(t) \cdot s(t) \quad \Rightarrow \quad W(2^i, 2^i n) = w^i_n \quad . \tag{4.29}$$

Define $\psi^g(t) \triangleq \sqrt{2}\, g(2t)$. Then the corresponding properties hold for $\mathbf{d}^i$ with g(t) replaced by $\psi^g(t)$; i.e.,

$$s^0_n = \phi_n(t) \cdot s(t) \quad \Rightarrow \quad d^i_n = \int (\bar{\psi}^g)^i_n\, (t)\, s(t) \quad . \tag{4.30}$$

$$\begin{aligned} s^0_n &= \phi_n \cdot s \text{ or} \\ s^0 &= s \text{ and } \mathbf{f} \text{ is } \grave{a}\ \textit{trous} \quad \Rightarrow \quad d^i_n = \sum_k (\psi^g)^i_n(k)\, s(k) \quad . \end{aligned} \tag{4.31}$$

The above results extend immediately to the undecimated transforms, $W(2^i, n)$, $w(2^i, n)$, and $\tilde{w}^i_n$ by translation invariance and Definition 3.1.

5. LAGRANGE INTERPOLATION FILTERS

A reasonable class of *ˋa trous* interpolators to consider for equation (2.10) are those that are exact for polynomials P(t) of degree $\leq$ M for some M, i.e., for which

$$\frac{1}{\sqrt{2}} P(\frac{n}{2}) = \sum_k f^{\dagger}_{n-2k} P(k) \quad . \tag{5.1}$$

For reasons that will become clear very shortly, we shall call these filters *Lagrange ˋa trous filters*. Since the *ˋa trous* filter **f** satisfies $f_{2k} = \delta_{0k}/\sqrt{2}$, equation (5.1) is an identity for n even. Let **a** contain the odd components of $\mathbf{f}^{\dagger}$,

$$a_k \triangleq \begin{cases} \sqrt{2}\, f^{\dagger}_{2k-1} & \text{for } k > 0 \\ \sqrt{2}\, f^{\dagger}_{2k+1} & \text{for } k < 0 \\ 1 & \text{for } k = 0 \end{cases} \quad . \tag{5.2}$$

Then (5.1) is equivalent to

$$P(\frac{n}{2}) = \sum_{k>0} a_k P(\frac{n+1}{2} - k) + \sum_{k<0} a_k P(\frac{n-1}{2} - k) \qquad \text{n odd} \tag{5.3}$$

for all polynomials P of degree $\leq$ M. We proceed to express the a_k in terms of Lagrange polynomials, and to show that the above conditions are essentially equivalent to $\mathbf{f} = \mathbf{h} * \mathbf{h}^{\dagger}$ where **h** is an appropriate Daubechies filter.

CONSTRUCTION OF THE FILTER a

First, we parameterize the family of filters **a** satisfying (5.3) by M+1, the dimension of the space of polynomials for which it must hold. For such a relationship to exist, one must relate the length of the filter (the number of unknowns) to M. To accomplish this, we shall assume that **a** has exactly the minimum number of coefficients needed to satisfy equation (5.3) . We further assume that **a** has symmetric support; i.e., there is an N such that $a_k = 0$ for $|k| > N$ and $a_k \neq 0$ for $|k| = N$. This assumption is not unreasonable, at least for symmetric wavelets g(t), since there is no a priori reason to distinguish between t and −t, and one would even expect **a** to be symmetric. We shall see, in fact, that the weaker condition of symmetric support joined with the previous constraints implies that **a** actually is symmetric.

The minimum number of coefficients (unknowns) equals the dimension of the space, i.e., satisfies 2N = M+1. Thus, the sums in (5.3) go from 1 to N. Moreover, since for any n, the polynomials $Q(x) \triangleq P(x - (n-1)/2)$ also form a basis, equation (5.3) is equivalent to

$$P(\frac{1}{2}) = \sum_{k=1}^{N} a_k P(1-k) + \sum_{k=-1}^{-N} a_k P(-k) \qquad \text{n odd} \quad , \tag{5.4}$$

which must hold for all P(x) of degree $\leq 2N-1$.

We pick out the kth coefficient by letting P be the Lagrange polynomial

$$L_j^{2N-1}(x) \triangleq \frac{\prod_{i\neq j}(x-i)}{\prod_{i\neq j}(j-i)} \qquad i, j \text{ in } [-N+1, N] \quad . \tag{5.5}$$

Then, $L_j^{2N-1}(k) = \delta_{jk}$, so that replacing P in equation (5.4) with L_j^{2N-1}, we get

$$a_j = L_{1-j}^{2N-1}(\frac{1}{2}) \qquad \text{for } j = 1, \ldots, N \tag{5.6a}$$

$$a_{-j} = L_j^{2N-1}(\frac{1}{2}) \qquad \text{for } j = 1, \ldots, N \quad . \tag{5.6b}$$

Inasmuch as the Lagrange polynomials L^{2N-1} form a basis for polynomials of degree less than or equal to $2N-1$, these a_k are in fact the unique solution to (5.4).

It is also straightforward to see from (5.5) and (5.6) that **a** is symmetric, i.e., for $j > 0$

$$a_j = \frac{\prod_{i\neq 1-j}(1/2-i)}{\prod_{i\neq 1-j}(-j+1-i)} = \frac{\prod_{i\neq j}(-1/2+i)}{\prod_{i\neq j}(-j+i)} = \frac{\prod_{i\neq j}(1/2-i)}{\prod_{i\neq j}(j-i)} = a_{-j} \tag{5.7}$$

In summary, we have

Theorem 5.1

Let **f** be an *a trous* filter, i.e.,

$$f_{2k} = \frac{1}{\sqrt{2}}\,\delta_{0k} \quad . \tag{5.8}$$

Assume, furthermore, that **f** is real with symmetric support described by $k \in [-2N+1, 2N-1]$. Then, **f** is a Lagrange *ˋa trous* filter, that is, (5.1) holds for all polynomials P of degree $\leq 2N-1$, if and only if the odd components of **f** are determined by equations (5.6). Furthermore **f** is necessarily symmetric.

RELATIONSHIP TO DAUBECHIES (QMF) FILTERS

In [8], Ingrid Daubechies constructs essentially the entire class of finite length filters **h** which satisfy (2.14) and fulfill suitable regularity conditions on (2.18). Explicitly, they take the form

$$\hat{h}(z) = \frac{1}{\sqrt{2}}\left(\frac{1}{2}(1+z)\right)^N Q(z) \tag{5.9}$$

where $\hat{h}(z) \triangleq \sum h_n z^{-n}$ is the z transform of **h**, and Q is an appropriately constrained polynomial. (In this section it is convenient to express the z transform as a polynomial which we denote $\hat{h}(z)$ where $h_z(\omega) = \hat{h}(e^{i\omega})$.) The derivation uses a specific **g**, which up to a phase factor is given by

$$g_n = (-1)^n h(1-n) \quad , \qquad (5.10)$$

and, as a consequence, (2.14a) reduces to

$$\Delta\, \mathbf{h}\, \mathbf{h}^{\dagger}\, D = I \quad . \qquad (5.11)$$

In other words, $[\mathbf{h}\mathbf{h}^{\dagger}]_{2n} = \delta_{0n}$, which is the *`a trous* condition for $\mathbf{h}\mathbf{h}^{\dagger}/\sqrt{2}$. Finally, if Q is taken to be of minimal degree (which turns out to be N − 1), $|Q|^2$ is unique. [8] In other words, the squares of these filters are characterized completely by satisfying (2.14d) and (5.11), and being of the form (5.9) with degree 2N − 1. We proceed to show that the $\mathbf{h} * \mathbf{h}^{\dagger}$ equal the Lagrange *`a trous* filters.

Let **h** be the Daubechies filter of order 2N. Since $\mathbf{h} * \mathbf{h}^{\dagger}$ is symmetric, the above conditions are equivalent to

$$\hat{\beta}(z) \triangleq \mathbf{h} * \mathbf{h}^{\dagger} = 1 + \sum_{1}^{N} b_k\, z^{2k-1} + \sum_{1}^{N} b_k\, z^{-2k+1} \qquad (5.12a)$$

and

$$\hat{\beta}(z) = \frac{1}{2}\,(\frac{1}{2}(1+z))^N\,(\frac{1}{2}(1+z^{-1}))^N\, Q(z)\, Q(z^{-1}) \quad . \qquad (5.12b)$$

That is, $\sqrt{2}\,\mathbf{f} = \mathbf{h} * \mathbf{h}^{\dagger}$ if and only if $\hat{f}(z)$ is of the form (5.12a) – (5.12b) and is of degree 2N − 1 in z (respectively, in z^{-1}).

Next, we show that the b_k coincide with the a_k of equation (5.3). We multiply (5.12a) by z^n for an arbitrary integer n,

$$z^n\, \hat{\beta}(z) = z^n + \sum_{1}^{N} b_k\, z^{n-1+2k} + \sum_{1}^{N} b_k\, z^{n+1-2k} \quad . \qquad (5.13)$$

The 2N zeros at z = −1 in (5.12b) imply that

$$\left.\frac{\partial^i (z^n \hat{\beta}(z))}{(\partial z)^i}\right|_{-1} = 0 \qquad i = 1, \ldots, 2N-1 \quad , \qquad (5.14a)$$

and since $\hat{\beta}(-1) = 0$, we also have

$$2\sum_{1}^{N} b_k = 1 \quad . \qquad (5.14b)$$

Next, we note that

$$\left.\frac{\partial^i(z^m)}{(\partial z)^i}\right|_{-1} = m\,(m-1)\cdots(m-i+1)\,(-1)^{m-i} \quad \text{for } i > 0. \qquad (5.15)$$

Define a set of polynomials P_i by

$$P_i(x) \triangleq 2x\,(2x-1)\cdots(2x-i+1) \quad . \qquad (5.16)$$

Finally, setting the i^{th} derivative of the right hand side of equation (5.13) at $z = -1$ to zero, using (5.15) and definition (5.16), we obtain

$$P_i(\frac{n}{2}) = \sum_{k>0} b_k\, P_i(\frac{n-1}{2} + k) + \sum_{k>0} b_k\, P_i(\frac{n+1}{2} - k) \qquad (5.17)$$

where $i = 1, ..., 2N-1$. The signs work out since $n-1+2k-i$ and $n+1+2k-i$ have the same parity while $n-i$ differs. Define $P_0(x) \equiv 1$. Then, since the polynomials $P_0(x)$ and $P_i(x)$ for $i = 1, ..., 2N-1$ form a basis for polynomials of degree $\leq 2N-1$, and since equation (5.14b) implies (5.17) for $i = 0$, equation (5.17) must hold for arbitrary polynomials $P(x)$ of degree $\leq 2N-1$. Replacing P_i by P in equation (5.17) and setting $a_k = b_{|k|}$ yields equation (5.3).

Conversely, Theorem 5.1 implies that if **a** satisfies equation (5.3) for all polynomials of degree $\leq 2N-1$ and has symmetric support, then it must be symmetric. Clearly (5.17) must be also be satisfied. From (5.2), this is equivalent to (5.12a) and (5.14) with $\sqrt{2}\,\hat{f}(z)$ replacing $\hat{\beta}(z)$ where $\hat{f}(z)$ is of degree $2N-1$ in z and the also in z^{-1}. Letting $n = 1$ (or 0) in (5.14a), we see that $\hat{f}(z)$ has 2N roots at $z = -1$. Since **f** is symmetric, $\hat{f}(z)$ must also have the form (5.12b). We conclude that $\sqrt{2}\,\mathbf{f} = \mathbf{h} * \mathbf{h}^\dagger$ where **h** is a Daubechies filter. Thus,

Theorem 5.2

There is a one-to-one correspondence between the squares of the Daubechies orthonormal wavelet filters **h** of length 2N and the Lagrange *'a trous* filters **f** of length 4N - 1 given by $\mathbf{f} = \mathbf{h} * \mathbf{h}^\dagger/\sqrt{2}$.

Note that one can compute the **h** of length 2N by taking all possible square roots of the Lagrange *'a trous* filters **f**.[10] **f** is easily computed from (5.6) where its even

[10] During the revision of this report, it was brought to the author's attention that an implicit relationship between the squared filters and Lagrange interpolation had been independently noted in private conversations between I. Daubechies and Ph. Tchamitchian.

components are given by $f_{2k} = \delta_{k0}/\sqrt{2}$, its odd positive components by $f_{2k-1} = b_k$ for k = 0 to N, and for odd negative k by symmetry. Also, the spectrum of **f**, $\hat{f}(e^{i\omega}) = |\hat{h}(e^{i\omega})|^2$, presents a convenient method of computing the power spectra of the **h**'s. In another vein, since the **h** are maximally flat filters (i.e., have the same number of vanishing derivatives at $z = 1$ and $z = -1$), Theorem 5.2 shows that a maximally flat filter is a Lagrangian interpolator, a fact which may aid in the design of such filters. [14]

6. WAVELET FILTERS IN SIGNAL PROCESSING

This section has its roots in a question which originally motivated the author to undertake this study: Inasmuch as the *`a trous* and Mallat algorithms share the same recursions, why not choose the Daubechies filters since they enjoy the additional advantage of orthonormality? The strongest arguments in favor of orthonormality seem to be mathematical elegance, ease of inversion, and, more subtly, good numerical properties. The major drawback is a lack of flexibility in filter design, in particular, an essentially fixed relative bandwidth. On the other hand, we have seen that the DWT has a firm analytical basis independent of the *`a trous* approximation, even in the nonorthogonal case. In the present section, we shall briefly examine the issues of inversion, boundedness, and adjustable relative bandwidth. (Another fundamental issue, regularity of the associated wavelet functions, appears in appendix B.) In particular, we wish to establish conditions under which these properties obtain, expressed directly in terms of the filters **f** and **g** rather than in terms, for example, of the scale function $\phi(t)$ or wavelet $\psi(t)$. This view is in the spirit of the DWT as an entity in its own right, and it is certainly a necessary element in deciding which filters to use in practice.

Any software realization of the wavelet transform only implements a finite number of octaves. Mathematically, this reduces inversion to an algebraic question, one of finding filters which satisfy certain (not overly restrictive) equations. However, other considerations begin to come into play. Exact inversion requires finite filters, and, even then, exceedingly long filters may not be useful. Moreover, the constrained problem is considerably more difficult to solve. An alternative approach, approximation by truncated infinite filters, might be acceptable, but, once again, practical considerations dictate that the filters decay quickly. Similarly, the behavior of the DWT at infinity (i.e., $\mathbf{w}^i$ as i goes to infinity) becomes relevant. For example, the condition $\sum_n g_n = 0$, the discrete counterpart of (1.2), is not necessary for inversion of a finite number of stages. However, it is necessary for finite energy and boundedness, which are desirable properties inasmuch as they reflect directly on the numerical stability of the algorithm and/or its inverse (see, for example, [3]).

INVERSION

To invert either the decimated or undecimated discrete wavelet transform[11] it suffices to invert a single stage (octave); that is, to find s^i, given s^{i+1} and w^{i+1} or $\tilde{w}^{i+1}$. The equations for inverting the decimated algorithm are exactly analogous to those for the Mallat algorithm pictured in figure 2.3. One seeks two filters **p** and **q** which invert a single stage of the decimated DWT in figure 1.1; i.e., such that

$$\begin{aligned} s^i &= \mathbf{PD}s^{i+1} + \mathbf{QD}s^{i+1} \\ &= (\mathbf{PD})(\underline{\triangle}\mathbf{F})\, s^i + (\mathbf{QD})(\underline{\triangle}\mathbf{G})\, s^i \ . \qquad (6.1) \end{aligned}$$

[11] More properly to invert a finite number of stages; see next subsection.

Equivalently,

$$(\mathbf{PD})(\Delta\mathbf{F}) + (\mathbf{QD})(\Delta\mathbf{G}) = \mathbf{I} \quad , \tag{6.2}$$

where I is the identity matrix. This type of equation, which in the frequency domain may be separated into two equations comparable to (2.14a) and (2.14b), has been treated extensively (but not exhaustively) in the subband coding literature (cf., [12], [13], or even [8]). The QMF filters of the Mallat algorithm satisfy (6.2) with $\mathbf{g}_z(\omega) = \mathbf{f}_z(\omega+\pi)$, $\mathbf{p} = \mathbf{f}^\dagger$, and $\mathbf{q} = \mathbf{g}^\dagger$ (i.e, equations (5.10) and (5.11)). A less restricted class is of the form $\mathbf{p} = \mathbf{f}^\dagger$ and $\mathbf{q} = \mathbf{g}^\dagger$. The general class of filters satisfying (6.2), so-called biorthogonal filters, are examined in [26] and [27]. It should be emphasized that for perfect reconstruction in applications all filters must be of finite length. This does not imply that infinite filters implemented by their truncations are not worthy of consideration. [26]

For the undecimated algorithm the requirements for inversion are much less stringent. In order to invert a stage of the algorithm of figure 3.2, the filters **p** and **q** need only satisfy (cf., figure 6.1)

$$\begin{aligned} \mathbf{s}^{i} &= \mathbf{p} * \mathbf{s}^{i+1} + \mathbf{q} * \tilde{\mathbf{w}}^{i+1} \\ &= (\mathbf{p} * \mathbf{f} + \mathbf{q} * \mathbf{g}) * \mathbf{s}^{i} \quad . \end{aligned} \tag{6.3}$$

That is,

$$\mathbf{p} * \mathbf{f} + \mathbf{q} * \mathbf{g} = \delta \tag{6.4}$$

where the Kronecker delta, $\delta \triangleq \delta_{0,m}$, is the identity for convolution. This is a single equation, and much more tractable than (6.2). If the polynomials formed by the z-transforms $\hat{\mathbf{f}}(z)$ and $\hat{\mathbf{g}}(z)$ are relatively prime, one may apply the Euclidean algorithm for the greatest common divisor (in this case, one) to find **p** and **q**. It has the advantage that finite **f** and **g** lead to finite **p** and **q**. Another method is simply to solve the equation in frequency space,

$$\mathbf{p}_z(\omega)\,\mathbf{f}_z(\omega) + \mathbf{q}_z(\omega)\,\mathbf{g}_z(\omega) = 1 \quad . \tag{6.5}$$

There is almost too much flexibility in solving this equation, although it becomes much more restrictive if one demands that the filters be finite or rapidly decreasing. Once again, a popular choice [28] is $\mathbf{f}_z\,\overline{\mathbf{f}_z} + \mathbf{g}_z\,\overline{\mathbf{g}_z} = 1$, which, for example, can be solved for $\mathbf{g}_z$ by taking the square root of $1 - |\mathbf{f}_z^2|$ as long as $|\mathbf{f}_z(\omega)| \leq 1$, (or, vice-versa, it can be solved for $\mathbf{f}_z$). The Daubechies (QMF) filters $\mathbf{h}/\surd 2$ and $\mathbf{g}/\surd 2$ certainly satisfy this equation so that inversion for the undecimated version of the Mallat algorithm is immediate. Another case, useful in signal processing, is to choose **f** to be *`a trous*, $\mathbf{p} = \mathbf{f}^\dagger/2$, and **g** any filter with nonvanishing spectrum except possibly where $|\mathbf{f}_z(\omega)|$ equals $\surd 2$ (cf., next subsection). Important questions of numerical stability, filter lengths, etc.,certainly remain to be answered, but are well beyond the scope of the present report.

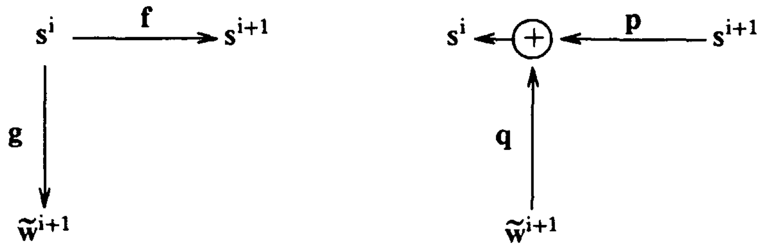

Figure 6.1 Illustration of a single stage and its inverse for the undecimated algorithm found in figure 3.1.

Finally, before departing from this subject, it should be mentioned that inversion of the undecimated case in the form of figure 3.1 also follows from (6.4). The inverting filters are just $\mathbf{D}^i\mathbf{p}$ and $\mathbf{D}^i\mathbf{q}$. It is a simple matter to verify that (6.4) implies that

$$(\mathbf{D}^i\mathbf{p}) * (\mathbf{D}^i\mathbf{f}) + (\mathbf{D}^i\mathbf{q}) * (\mathbf{D}^i\mathbf{g}) = \delta \quad . \tag{6.6}$$

(Inserting zeros in δ just yields δ.)

FINITE ENERGY AND BOUNDEDNESS

The discrete wavelet transform is a mapping of sequences s_n, $n = 1, 2, \ldots$ into the space of doubly indexed sequences w_n^i, $i,n = 1, 2, \ldots$ Finite energy for the signal is simply

$$\sum_n |s_n|^2 < \infty \quad . \tag{6.7}$$

One's initial tendency is to look for the same condition on the wavelet transform, i.e., $\sum_{i,n} |w_n^i|^2 < \infty$. However, the actual situation is not quite so transparent. Referring back to the continuous case, we have [16]

$$\int |s(t)|^2 \, dt = \int |W(a, b)|^2 \, \frac{da\, db}{a^2} \quad , \tag{6.8}$$

i.e., the weight (measure) $da\, db / a^2$ results in units of energy. For the DWT, $a = 2^i$ and db is either 1 (the undecimated case) or 2^i (the decimated case). Discretizing and noting that $da / a = d(\ln a)$, we see that

$$\frac{da\,db}{a^2} \rightarrow \frac{(\ln 2)\,\Delta i\,\Delta b}{2^i} = \ln 2\,(\frac{\Delta b}{2^i}) \qquad (6.9)$$

since $\Delta i = 1$. Observing that octave i has a bandwidth, up to a constant factor, of $1/2^i$, we assign (6.9) the following physical interpretation: The discrete wavelet transform is in units of power/Hz. Multiplying by the bandwidth $1/2^i$ gives power, and additional multiplication by the time interval Δb gives energy. Note that the decimated version only outputs every 2^i points so that in that case $\Delta b = 2^i$. In summary,

(i) $|w_n^i|^2$, $|\tilde{w}_n^i|^2$ are in units of power/Hz.

(ii) Octave i has bandwidth $\approx 1/2^i$.

(iii) $|\tilde{w}_n^i|^2/2^i$ and $|w_n^i|^2$ are in units of energy.

One should take care to note that for w_n^i, the energy weight $da\,db/a^2$ is a constant independent of i so that $|w_n^i|^2$ is discretized in a fashion so as to be both power/Hz and energy.

Let $||s||^2 \triangleq \sum_n |s_n|^2$ be the squared norm of s, and define

$$||\tilde{s}^\infty||^2 \triangleq \lim_{i\rightarrow\infty} \sup \frac{1}{2^i} ||s^i||^2 \ , \qquad (6.10)$$

which corresponds to the DC energy (i.e., at $\omega = 0$). Finally, define the energy of the DWT by

$$\tilde{E} = \sum_i \frac{1}{2^i} ||\tilde{w}^i||^2 + ||\tilde{s}^\infty||^2 \qquad (6.11a)$$

$$E = \sum_i ||w^i||^2 + ||s^\infty||^2 \ . \qquad (6.11b)$$

Energy conservation takes the form of the following Parseval's relationship for discrete wavelets:

Definition 6.1

A particular choice of filters **f** and **g** is said to be energy conserving if

$$||s||^2 = C\,(\sum_i \frac{1}{2^i} ||\tilde{w}^i||^2 + ||\tilde{s}^\infty||^2) \ . \qquad (6.12)$$

One may also specify conservation for decimated transforms, in which case the 2^i is dropped.

Except for some clarifying remarks at the end, we restrict the discussion in the remainder of this subsection to the undecimated DWT. Following the above definitions, we see that the wavelet transform will have finite energy if and only if

$$\sum_i \frac{1}{2^i} ||\tilde{w}^i||^2 + ||\tilde{s}^\infty||^2 < \infty \ . \tag{6.13}$$

This is a necessary condition for the mapping $\tilde{w}$, from $l^2(Z)$ to $l^2(Z^2; 2^{-i}, 1)$ to be bounded. Of course, in practice, implementations never compute an infinite number of octaves. Nevertheless, the property (6.13) of finite energy can be quite important. Unbounded transformations tend to have poor numerical behavior even when truncated. Similarly, the inverse will not be bounded unless the series (6.13) is bounded below. A wavelet representation which has these properties,

$$A\,||s||^2 \leq \sum_i \frac{1}{2^i} ||\tilde{w}^i||^2 + ||\tilde{s}^\infty||^2 \leq B\,||s||^2 \ , \tag{6.14}$$

is called a frame (cf., [3], although here the ambient Hilbert space is l^2 rather than $L^2(R)$.) We proceed to derive conditions on the filters **f** and **g** for (6.14) to hold.

Figure 3.1 implies that

$$\int_{-\pi}^{\pi} |s_z^{i+1}(\omega)|^2 + \int_{-\pi}^{\pi} |\tilde{w}_z^{i+1}(\omega)|^2 = \int_{-\pi}^{\pi} (|\mathbf{f}_z(2^i\omega)|^2 + |\mathbf{g}_z(2^i\omega)|^2)\, |s_z^i(\omega)|^2 \ . \tag{6.15}$$

Suppose that

$$\max_\omega \frac{1}{2}(|\mathbf{f}_z(\omega)|^2 + |\mathbf{g}_z(\omega)|^2) \leq 1 \ . \tag{6.16}$$

Then, in the time domain, (6.15) and (6.16) imply

$$\frac{1}{2^{i+1}} ||s^{i+1}||^2 + \frac{1}{2^{i+1}} ||\tilde{w}^{i+1}||^2 \leq \frac{1}{2^i} ||s^i||^2 \ . \tag{6.17}$$

Adding $||\tilde{w}^i||^2/2^i$ to both sides and repeating for decreasing octaves implies that

$$\frac{1}{2^{J+1}} ||s^{J+1}||^2 + \sum_{i\leq J} \frac{1}{2^i} ||\tilde{w}^i||^2 \leq ||s||^2 \ . \tag{6.18}$$

Finally, letting J go to infinity, we get not only (6.13), but also the right inequality of (6.14) with $B = 1$.

However, the condition (6.16) is much too strong. That is, the transformation $\mathbf{g} \rightarrow C\,\mathbf{g}$ for a large enough constant C would cause (6.16) to be violated even though C has no effect other than to multiply the total energy by a constant. In fact the filters **f** and **g** produce finite energy transforms if and only if **f** and C**g** yield finite energy. Thus, to have finite energy, it is sufficient to find a $C > 0$ such that $\max_{\omega} (|\mathbf{f}_z(\omega)|^2 + C\ |\mathbf{g}_z(\omega)|^2) \leq 2$. Such a C exists provided that $|\mathbf{f}_z(\omega)|^2 \leq 2$ and $(1/2)\ |\mathbf{g}_z(\omega)|^2/(1 - \frac{1}{2}\ |\mathbf{f}_z(\omega)|^2)$ is finite; i.e., is less than some finite $B = 1/C$. A similar argument holds for the lower bound. If

$$\min_{\omega} \frac{1}{2}(|\mathbf{f}_z(\omega)|^2 + |\mathbf{g}_z(\omega)|^2) \geq 1 \quad , \tag{6.19}$$

then equation (6.18) holds with the inequality reversed and the left inequality of (6.14) holds with $A = 1$. Once again, we apply the trick with the constant C and find that, for the inverse to be bounded, it is sufficient that there exist $A = 1/C > 0$ such that $(1/2)\ |\mathbf{g}_z(\omega)|^2/(1 - \frac{1}{2}\ |\mathbf{f}_z(\omega)|^2) \geq A$. In summary,

Theorem 6.1

A sufficient condition for the undecimated DFT $\tilde{\mathbf{w}}$ and its inverse to satisfy (6.14) (that is, to be bounded) is that, for all ω, $|\mathbf{f}_z(\omega)|^2 \leq 2$ and

$$0 < A \leq \frac{\frac{1}{2}\ |\mathbf{g}_z(\omega)|^2}{1 - \frac{1}{2}|\mathbf{f}_z(\omega)|^2} \leq B < \infty \quad . \tag{6.20}$$

To satisfy (6.20), one must have $|\mathbf{f}_z(\omega)| = 2 \iff \mathbf{g}_z(\omega) = 0$, and the multiplicities of the corresponding roots must be identical. Note, also, that (6.20) can be used to give an estimate of B/A, the so-called tightness of the frame.

Whether these conditions are also necessary remains an open question. One can, however, show from equation (3.11) and an examination of the power $\tilde{\mathbf{w}}_z^i(\omega)$ at $\omega = 0$ that a necessary condition is $\mathbf{g}_z(0) = 0$ (equivalently, $\sum_n g_n = 0$). This is the discrete analogue of the admissibility condition (1.2). The author conjectures that in the discrete case it is not a sufficient condition. (We remind the reader that even in those cases for which the DWT is exactly the sampled WT, finite energy of the continuous wavelet transform does not imply that of the discrete transform.) We do have, however,

Theorem 6.2

A necessary and sufficient condition for energy conservation (equation (6.12)) is that for all ω

$$\frac{1}{2}(|\mathbf{f}_z(\omega)|^2 + \frac{1}{C}\ |\mathbf{g}_z(\omega)|^2) = 1 \quad . \tag{6.21}$$

To prove this, we may, without loss of generality, set $C = 1$ (i.e., redefine $\mathbf{g}$ by the constant factor C). Sufficiency follows as above, with inequalities replaced by equalities. To prove necessity, we first note that energy preservation for arbitrary signals implies the energy must be conserved for each stage. (For example, if the signal is s^1, energy must be preserved, and since it is preserved for s^0, the first stage must preserve energy.) From (6.15) this implies (6.21) with C equal to one.

The decimated case seems to present problems. For the above proofs to carry over, the even part of the signal would need to compensate for the lack of the factor $1/2^i$ in (6.11b), but $||s^i_{\text{even}}||^2 \neq 1/2\,||s^i||^2$. This problem presents yet another area for additional research.

RESOLUTION AND RELATIVE BANDWIDTH

Considerable insight may be gained by viewing the algorithm in the frequency domain. One stage of the decimated DWT, illustrating equations (4.11) from this point of view, is pictured in figure 6.2. Since we are dealing with the discrete wavelet transforms, $\hat{s}(z) = \hat{s}(e^{i\omega})$ is evaluated on the unit circle. For convenience only the positive frequencies are pictured. Briefly, the algorithm is

(a) Bandpass the upper half of the spectrum to yield $\mathbf{w}^i$.
(b) Lowpass to obtain the lower half of the spectrum ($[0, \pi/2]$).
(c) Decimate to expand the lower half to $[0, \pi]$.
(d) Go to (a).

In somewhat more detail: We first obtain the high-frequency information by using $\mathbf{g}$ to filter the upper half of the spectrum of $\mathbf{s}^i$. The filter output is $\mathbf{w}^i$. Then, in preparation for the next octave, $\mathbf{s}^i$ is lowpass filtered by $\mathbf{f}$. This retains the, as yet unexamined, low-frequency contents and also prevents the upper half of the spectrum from aliasing (i.e., contaminating the low-frequency contents) in the dilation which follows. Finally, the operator $\hat{\Delta}$ spreads[12] the remaining energy to fill the spectrum, producing octave i + 1. The procedure then repeats itself, s^{i+1} is bandpass filtered to get the spectral contents at frequencies which are, in absolute units, one-half the frequencies of the previous octave.

A potential problem is immediately apparent. If the bandwidth of $\mathbf{g}_z(\omega)$ is less than $\pi/2$, a portion of the signal energy will be discarded; it never appears in $\mathbf{w}^i$. One possible remedy is to make $\mathbf{g}_z$ sufficiently broad; however, that would limit the resolution. Alternatively, we may introduce so-called *voices*. That is, we can employ a bank of filters of the type $\mathbf{g}$ (see figure 6.3) in order to cover the entire upper half of the spectrum.

[12] Δ, which decimates, is a contraction. Its Fourier transform $\hat{\Delta}$ is thus a dilation.

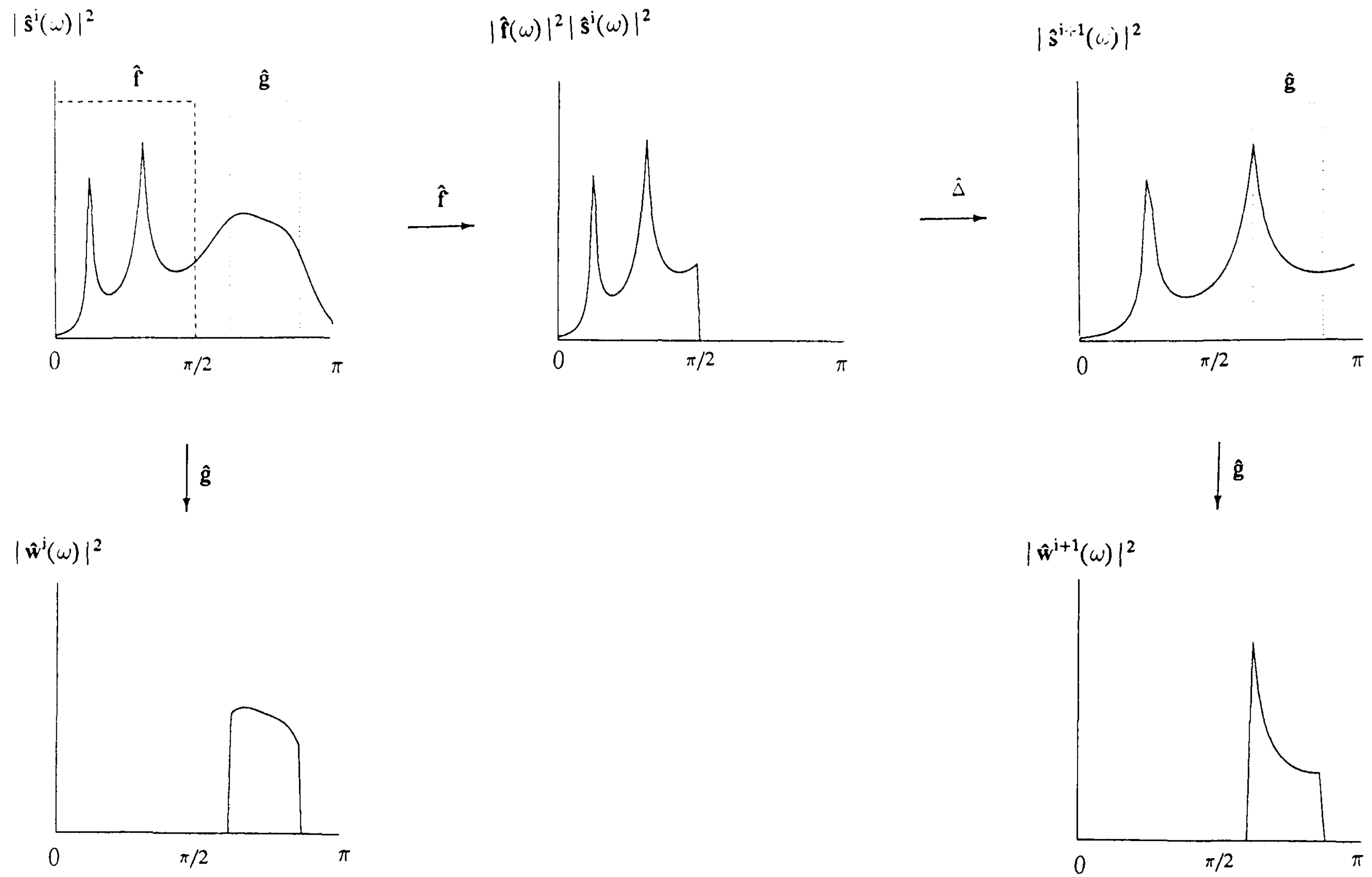

Figure 6.2 Illustraticn of one stage of the algorithm, octaves i and i + 1 inclusive, viewed from the frequency domain.

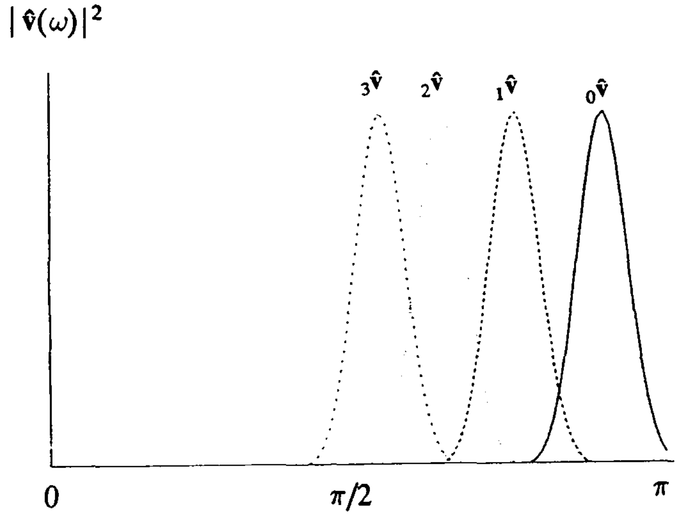

Figure 6.3 Plot of the power spectra of the bandpass filters for four voices where $\hat{g}(\omega)$ is given by (6.23).

We formalize some of these concepts using the modulated Gaussian of (1.4) as an example. With the introduction of an additional parameter β, g(t) becomes

$$g(t) \triangleq e^{i\nu t}\, e^{-\beta^2 t^2/2} \quad . \tag{6.22}$$

Its Fourier transform is given by

$$\hat{g}(\omega) = \frac{1}{\beta}\, e^{-(\omega - \nu)^2/2\beta^2} \quad . \tag{6.23}$$

We define the bandwidth of $\hat{g}(\omega)$ as twice the interval between points for which the modulus of (6.23) drops to 1/e of its peak value, i.e.,

$$\text{BW} \triangleq 2\sqrt{2}\beta \ . \tag{6.24}$$

The filter **g** is the sampled version of (6.22), that is,

$$g_n \triangleq e^{i\nu n}\, e^{-\beta^2 n^2/2} \quad . \tag{6.25}$$

For convenience, we set the sample rate equal to one. The following three restrictions on ν and β are necessary: First, in order that $\mathbf{g}_z(\omega)$ lie in the upper half of the spectrum (cf., figure 6.2), we require that

$$\frac{\pi}{2} \leq \nu \quad . \tag{6.26a}$$

Next, in order that g(t) be admissible and analytic (cf., [18]), we demand

$$\beta \leq \frac{\nu}{2\pi} \quad . \tag{6.26b}$$

Under (6.26b), $\mathbf{g}_z(\omega) \sim 0$ for $\omega \leq 0$. ([18] recommends $\beta \leq \nu/5$ as being sufficient.) Finally, in order that the spectrum not be aliased, we set

$$\nu \leq \pi - \sqrt{2}\beta \ . \tag{6.26c}$$

These may be summarized in

$$\max(2\pi\beta, \pi/2) \leq \nu \leq \pi - \sqrt{2}\beta \quad . \tag{6.27}$$

At this point a word of caution is advised. The bandwidth of the discrete filter **g** (e.g., (6.25)) is $2\sqrt{2}\,\beta$ only when the sample rate is 1. Since the ith octave is the result of i decimations by 2, sampling the original signal s^0 at a rate $\Delta t = 1$ results in a Nyquist frequency of $\pi/2^i$ for octave i (i.e., for s^i). Thus, the central frequency of **g**, considered as a filter on octave i is $\nu/2^i$, and its bandwidth is $2\sqrt{2}\beta/2^i$. For this reason, it is simpler and less ambiguous to speak in terms of a relative bandwidth which is independent of sample rate. More precisely, we define

$$\text{RBW} \triangleq \frac{\text{BW}}{\text{mean frequency}} \quad . \tag{6.28}$$

(Any appropriate representative of the "center" of the filter can replace the mean frequency in (6.28).) In the case of **g**, we have

$$\text{RBW} = \frac{(2\sqrt{2}\beta/2^i)}{(\nu/2^i)} = \frac{2\sqrt{2}\beta}{\nu} \quad . \tag{6.29}$$

Also, since $\pi/2 \leq \nu \leq \pi$, we have $2\sqrt{2}\beta/\pi \leq \text{RBW} \leq 4\sqrt{2}\beta/\pi$, or approximately

$$\beta \leq \text{RBW} \leq 2\beta \ . \tag{6.30}$$

In view of (6.30), we shall consider the parameter β as, essentially, the relative bandwidth of **g**.

The number of voices M that we would expect to need to cover the upper half of the spectrum is

$$M \sim \frac{\pi/2}{2\sqrt{2}\beta} \sim \frac{1}{2\beta} \ . \tag{6.31}$$

The filter for voice j, which we denote ${}_j\mathbf{v}$, is determined by sampiing the function $g(t/\alpha^j)$, i.e.,

$$({}_j\mathbf{v})_n \triangleq g(n/\alpha^j) \quad \text{for } j = 0 ,\ldots, M-1 \text{ and } \alpha \triangleq 2^{1/M} \ . \tag{6.32}$$

The power spectra of the ${}_j\mathbf{v}$ are illustrated in figure 6.3. An alternative would be to define the voices as frequency translations of the filter **g**. However, (6.32) seems to be a more natural definition since it maintains the affine structure. (It is equivalent to taking a nondyadic value for the dilation parameter *a* in equation (1.1).) Note that the bandwidths of the voices decrease with j, and the spectral spacing $2\sqrt{2}\beta/\alpha^j$ differs somewhat from that assumed in (6.31).

Control over the relative resolution is important because, although the absolute resolution $(\beta/2^i)$ improves at higher octaves, at the same time, the analysis frequency decreases. If one wishes to improve resolution at a given frequency, one has to better the relative resolution, i.e., decrease β. On the other hand, the standard tradeoffs apply to choosing β. Small β increases the relative resolution but also requires more voices and a longer filter **g**; thus, more computation. (The duration of g(t) is on the order of $2\sqrt{2}/\beta$ so that to provide a reasonable approximation to (6.22), the length of **g** must be proportional to $1/\beta$.) Moreover, the increased length of **g** implies a worsening of the (relative) time resolution. The time-bandwidth product is bounded below by the uncertainty principle, and no amount of computation will simultaneously produce arbitrarily small time and frequency resolution in a single wavelet transform.

With respect to choosing the lowpass filter **f**, we note that the spectrum of a longer filter **f** will generally have a a sharper cutoff. This cutoff is relevant because it prevents the energy in the upper half of the spectrum from leaking (aliasing) into the iower half under the decimation, $\hat{\Delta}$. For most applications, a Lagrange *`a trous* filter of length 7 (N = 2) is sufficient. [5] One can, of course, also use the asymmetric filters **h**. Their discrimination of temporal direction seems intriguing, but remains uninvestigated.

Finally, how do the above considerations relate to orthonormal wavelets? The power spectra of filters **h** and **g** obeying (2.14) satisfy (cf., [8], [20])

$$|\mathbf{h}_z(\omega)|^2 + |\mathbf{g}_z(\omega)|^2 = 2 \tag{6.33a}$$

and

$$|\mathbf{h}_z(0)| = |\mathbf{g}_z(\pm\pi)| = \sqrt{2} \ . \tag{6.33b}$$

It follows that, for positive frequencies (likewise, for negative frequencies), **g** and **h** must each maintain a bandwidth on the order of $\pi/2$.[13] Thus, in exchange for orthonormality one relinquishes control over bandwidth. The relative bandwidth is, essentially, fixed at $(\pi/2) \; / \; (3\pi/4) = 2/3$. On the other hand, if one wishes to give time and frequency localization equal weight (e.g., $\beta = 1$ so that bandwidth = duration = $2\sqrt{2}$), a relative bandwidth in the neighborhood of 2/3 is in a sense optimal.

[13] The larger the value of N, the more rapid the asymptotic convergence of $\hat{\psi}(\omega)$ to zero; i.e., as $\omega \rightarrow \infty$. [8] This hints at a smaller bandwidth for $\hat{\psi}$ and, hence, also for $\mathbf{g}_z$ for large N. However, the speed at which the $\hat{\psi}(\omega)$ fall off near $\omega = 0$ appears to be fairly insensitive to N.

7. CONCLUSION

We have seen that the `*a trous* algorithm bears an intimate relationship to Mallat's multiresolution algorithm. Originally devised as a computationally efficient implementation, it is more properly viewed as a nonorthogonal multiresolution algorithm for which the discrete wavelet transform is exact. Moreover, the commonly used Lagrange `*a trous* filters are simply the convolutional squares of the Daubechies filters for compact orthonormal wavelets.

From a broader viewpoint, these two algorithms are instances of the discrete wavelet transform (DWT), which, in more conventional terms, is simply a filter bank utilizing decimation and two filters. There are two basic versions of the DWT, one of which is simply the decimated output (octave i is decimated by 2^i) of the other. The decimated DWT is characterized by octaves: (a) obtained by alternating a lowpass filter **f** with decimation, and (b) tapped by a bandpass filter **g** to produce the output. The undecimated DWT inserts i zeros between the elements of the filters at octave i in lieu of decimation. (In the case of voices, several **g**'s are used.) Finally, we note that under very general conditions, there exists a function $\psi^g(t)$ such that the fil'er bank outputs w_n^i correspond to :he sampled wavelet transform $\frac{1}{\sqrt{2^i}} \int \bar{\psi}^g(\frac{t}{2^i} - n)\, s(t)\, dt$, thus, justifying the terminology discrete wavelet transform.

The personality of a given DWT is distinguished by the choice of filters. If **f** satisfies the `*a trous* condition $f_{2n} = \delta_{n0}/\sqrt{2}$, then **g** is the sampled version of $\psi^g(t)$; i.e., $g_n = \psi^g(n)$. If finite length **f** and **g** obey the constraints of the multiresolution algorithm, then the $\sqrt{2^i}(\psi^g)(2^i t - n)$ are the compact orthonormal wavelets. A number of fundamental constraints have been discussed. In various combinations they have a bearing on the regularity of the wavelet function, on the energy in the transform domain, and on the boundedness and invertibility of the transform. In particular, we have provided a set of conditions on the filters sufficient for the transform and its inverse to be bounded. The signal-processing properties of the discrete wavelet transform depends particularly strongly on the choice of **g**. The general constraints mentioned above are not very restrictive on **g**; however, there is considerably less freedom in the orthonormal case. In particular, if orthonormality is a requirement, the half bandwidth of **g** (and, hence, the relative bandwidth of the wavelet) is no longer adjustable. It remains fixed at approximately $\pi/2$.

Many topics remain for investigation. Although considerable work has been done in finding filter pairs which have a complementary set for the inverse transform (cf., [8], [26], [27]), it is far from exhaustive. The equivalence of maximally flat filters (with equal order roots at 0 and π) with Lagrange `*a trous* filters as a design tool is perhaps worthy of investigation. Many of our filter conditions on energy are sufficient but possibly not necessary; a tight set of necessary and sufficient conditions for boundedness would be desirable. Finally, an investigation into the quality of the approximation of the DWT to the sampled WT in the case where it is not exact could be fruitful. It would perhaps lend more insight into the role of the regularity of the wavelet function in particular applications.

8. REFERENCES

1. *Wavelets: Time-Frequency Methods in Phase Space*, Proceedings of the International Conference, Marseille, Dec 1987, J. M. Combes, A. Grossman, Ph. Tchamitchian eds., Springer-Verlag, 1989.

2. Grossmann, A., R. Kronland-Martinet, and J. Morlet. "Analysis of Sound Patterns Through Wavelet Transforms," *Int. J. of Pattern Recognition and Artif. Int.*, 1, Aug 1987, pp 97-126.

3. Daubechies, I. "The Wavelet Transform, Time-Frequency Localization, and Signal Analysis," *IEEE Trans. on Information Theory*, to appear.

4. Holschneider, M., R. Kronland-Martinet, J. Morlet, and Ph. Tchamitchian. "A Real-Time Algorithm for Signal Analysis with the Help of the Wavelet Transform," reference [1] above, pp 286-297.

5. Dutilleux, P. "An Implementation of the *Algorithme 'a Trous* to Compute the Wavelet Transform," reference [1] above, pp 298-304.

6. Mallat, S. G. "A Theory for Multiresolution Signal Decomposition: the Wavelet Representation," *IEEE Trans. on Pattern Analysis and Machine Int.*, 2, Jul 1989, pp 674-693.

7. Mallat, S. G. "Multifrequency Channel Decompositions of Images and Wavelet Models," *IEEE Trans. ASSP*, 37, Dec 1989, pp 2091-2110.

8. Daubechies, I. "Orthonormal Bases of Compactly Supported Wavelets," *Comm. Pure Appl. Math.*, 41, 1988, pp. 909-996.

9. Daubechies, I. "Orthonormal Bases of Wavelets with Compact Support – Connection with Discrete Filters," reference [1] above, pp 38-66.

10. Meyer, Y. "Ondelettes, Fonction Splines, et Analyses Gradue'es," Univ. of Torino, 1986.

11. Esteban, D. and C. Galand. "Application of Quadrature Mirror Filters to Split-Band Coding," *Proc. ICASSP*, May 1977, pp 191-195.

12. Vetterli, M. "A Theory of Multirate Filter Banks," *IEEE Trans. ASSP*, 35, March 1987, pp 356-372.

13. Smith, M. J. T. and T. P. Barnwell III. "Exact Reconstruction Techniques for Tree-Structured Subband Coders," *IEEE Trans. ASSP*, 34, Jun 1986, pp 434-441.

14. Hermann, O. "On the Approximation Problem in Nonrecursive Digital Filter Design," *IEEE Trans. C.T.*, 18, May 1971, pp 411-413.

15. Heil, C. and D. Walnut, "Continuous and Discrete Wavelet Transforms," *SIAM Review*, 31, Dec 1989, pp 628-666.

16. Grossman, A. and J. Morlet, "Decomposition of Hardy functions into Square Integrable Wavelets of Constant Shape," *SIAM J. Math. Anal.*, 15, 1984, pp 723-736.

17. Morlet, J., G. Arens, E. Fourgeau, D. Giard. "Wave Propagation and Sampling Theory," *Geophysics*, 47, Feb 1982, pp 203-236.

18. Grossmann, A., R. Kronland-Martinet, and J. Morlet. "Reading and Understanding Continuous Wavelet Transforms," reference [1] above, pp 2-20.

19. Daubechies, I. "Orthonormal Bases of Compactly Supported Wavelets II: Variations on a Theme," preprint.

20. Shensa, M. J. "Implementing Affine Wavelet Transforms in Signal Processing," NOSC Tech. Report, in preparation.

21. Gambardella, G. "A Contribution to the Theory of Short-Time Spectral Analysis with Nonuniform Bandwidth Filters," *IEEE Trans. Circuit Theory*, 18, Jul 1971, pp 455-460.

22. Altes, R. "Sonar for Generalized Target Description and its Similarity to Animal Echolocation Systems," *J. Acoust. Soc. Am.*, 59, Jan 1976, pp 97-206.

23. Vetterli, M. and C. Herley. "Wavelets and Filter Banks: Relationships and New Results," *Proc. ICASSP*, Apr 1990, pp 1723-1726.

24. Kumar, A. and D. R. Fuhrmann. "The Frazier-Jawerth Transform," *Proc. ICASSP*, Apr 1990, pp 2483-2486.

25. Mallat, S. "Multiresolution Approximations and Wavelet Orthonormal Bases of $L^2(R)$," *Trans. AMS*, 315, Sept 1989, pp 69-87.

26. Vetterli, M. and C. Herley. "Wavelets and Filter Banks: Theory and Design," submitted to *IEEE Trans. Signal Processing*, 1990.

27. Cohen, A., I. Daubechies, and J-C. Feauveau. "Biorthogonal Bases of Compactly Supported Wavelets," in press.

28. Mallat, S. G. and S. Zhong. "Complete Signal Representation with Multiscale Edges," NYU Computer Science Tech. Report 483, Dec 1989.

APPENDIX A – PROOF OF LEMMA 3.1 AND LEMMA 3.2

Lemma 3.1

$$(\Delta \mathbf{F})^{i}{}_{mk} = (\Delta \mathbf{F})^{i}{}_{0,\, k-2^{i}m} \tag{A.1}$$

Proof

For i = 1, we have

$$(\Delta \mathbf{F})_{mk} = f_{2m-k} = (\Delta \mathbf{F})_{0,\, k-2m} \quad . \tag{A.2}$$

Then, by induction,

$$\begin{aligned}
(\Delta \mathbf{F})^{i}{}_{mk} &= \sum_{j} f_{2m-j}\ (\Delta \mathbf{F})^{i-1}{}_{jk} \\
&= \sum_{j} f_{2m-j}\ (\Delta \mathbf{F})^{i-1}{}_{0,k-2^{i-1}j} \\
&= \sum_{j} f_{-j}\ (\Delta \mathbf{F})^{i-1}{}_{0,k-2^{i-1}j-2^{i}m} \\
&= \sum_{j} f_{-j}\ (\Delta \mathbf{F})^{i-1}{}_{j,k-2^{i}m} \\
&= (\Delta \mathbf{F})^{i}{}_{0,k-2^{i}m}
\end{aligned} \tag{A.3}$$

Lemma 3.2

$$\sum_{k} (\Delta \mathbf{F})^{i}{}_{nk}\, e^{ik\omega} = e^{i2^{i}n\omega} \prod_{j=0}^{i-1} \mathbf{f}_{z}(2^{j}\omega) \quad . \tag{A.4}$$

Proof

For i = 1, we have

$$\begin{aligned}
\sum_{k} (\Delta \mathbf{F})_{nk}\, e^{ik\omega} &= \sum_{k} f_{2n-k}\, e^{ik\omega} \\
&= \sum_{k} e^{i2n\omega}\, \mathbf{f}_{z}(\omega)
\end{aligned} \tag{A.5}$$

Then, by induction,

$$\begin{aligned}
\sum_{k} (\Delta \mathbf{F})^{i}{}_{nk}\, e^{ik\omega} &= \sum_{k} (\Delta \mathbf{F})_{nk}\, e^{ik(2^{i-1}\omega)} \prod_{j=0}^{i-2} \mathbf{f}_{z}(2^{j}\omega) \quad . \\
&= e^{i2^{i}n\omega} \prod_{j=0}^{i-1} \mathbf{f}_{z}(2^{j}\omega) \quad .
\end{aligned} \tag{A.6}$$

APPENDIX B – SUMMARY OF FILTER CONSTRAINTS

The discrete wavelet transform $\tilde{\mathbf{w}}^i$ and the decimated discrete wavelet transform $\mathbf{w}^i$ (or $\mathbf{d}^i$) are defined for arbitrary filters $\mathbf{f}$ and $\mathbf{g}$ by

$$s^{i+1} = (\mathbf{D}^i\mathbf{f}) * s^i \qquad \text{(B.1a)}$$

$$\tilde{w}^i = (\mathbf{D}^i\mathbf{g}) * s^i \quad , \qquad \text{(B.1b)}$$

and

$$s^{i+1} = \Delta\,(\mathbf{f} * s^i) \qquad \text{(B.2a)}$$

$$w^i = \mathbf{g} * s^i \qquad \text{(B.2b)}$$

$$\mathbf{d}^{i+1} \triangleq \Delta\, w^i \quad , \qquad \text{(B.2c)}$$

respectively. Also,

$$w^i_n = \tilde{w}^i_{2^i n} \; . \qquad \text{(B.3)}$$

In essentially all applications $\mathbf{f}$ is a lowpass filter and $\mathbf{g}$ is highpass. This rather vague ,ualification is quantified below.

That is, in addition to the above definitions, it is expedient to impose auxiliary conditions on the filters to insure (a) that the DWT is related to some WT with a reasonably behaved scale function $\phi(t)$; (b) that the transform have finite energy and be a bounded transformation; and, often, (c) that it be invertible. The algebraic conditions for invertibility are found in equations (6.2) and (6.4). At the time of this report, no single set of necessary and sufficient conditions exist for the satisfaction of (a) and (b). Indeed, the definition of "reasonable behavior" of the scale function ultimately depends on the application. In an attempt to provide some degree of organization, we first list the candidate constraints, loosely labeled as lowpass, highpass, or energy conditions. We then summarize their consequences. If either of the filters is infinite it is assumed to satisfy the decay condition (cf., [8], [25])

$$\exists\, \epsilon \text{ such that } \sum_n |f_n| n^\epsilon < \infty \; . \qquad \text{(B.4)}$$

CANDIDATE CONSTRAINTS

(i) lowpass

$$\sum_n f_n = \sqrt{2} \qquad \text{(B.5)}$$

$$\left(\text{i.e., } \frac{1}{2}\mathbf{f}_z(0) = 1\right)$$

(ii) energy

$$\frac{1}{2}|\mathbf{f}_z(\omega)|^2 \leq 1 \tag{B.6}$$

(iii) lowpass

$$\frac{1}{\sqrt{2}}\mathbf{f}_z(\omega) = (1 + e^{i\omega})^N \gamma(\omega) \tag{B.7}$$

$$\text{where } |\gamma(\omega)| \leq C < 1$$

(Note that (B.7) implies that $\mathbf{f}_z(\pi) = 0$)

(iv) energy: complementary lowpass/highpass pair

$$0 < A \leq \frac{\frac{1}{2}|\mathbf{g}_z(\omega)|^2}{1 - \frac{1}{2}|\mathbf{f}_z(\omega)|^2} \leq B < \infty \ . \tag{B.8}$$

Equation (B.8) implies

(a) highpass

$$\sum_n g_n = 0 \tag{B.9}$$

(i.e., $\mathbf{g}_z(0) = 0$)

(b) lowpass/highpass

$$\frac{1}{2}|\mathbf{f}_z(\omega)| = 1 \Rightarrow \mathbf{g}_z(\omega) = 0 \tag{B.10a}$$

$$\mathbf{g}_z(\omega) = 0 \Rightarrow \frac{1}{2}|\mathbf{f}_z(\omega)| = 1 \tag{B.10b}$$

(v) energy

$$\frac{1}{2}(|\mathbf{f}_z(\omega)|^2 + C\,|\mathbf{g}_z(\omega)|^2) = 1 \ . \tag{B.11}$$

IMPLICATIONS

Necessity for pointwise convergence of (4.1) to $\hat{\phi}(\omega)$: (B.5)

Sufficiency for (4.1) and (4.4) to converge in $L^1(R)$, and $L^2(R)$ to continuous $\phi(t)$ and $\hat{\phi}(\omega)$ respectively: (B.5), (B.6), and (B.7).

This is one of the central results of [8], which also includes an examination of the decay of $\phi(t)$ and other regularity properties. Note, that an important class of wavelets which do not fall under the domain of this theorem is the Haar wavelets (cf.,equation (2.28a)). Pointwise convergence still holds for the Haar wavelets, but they are not continuous.

Necessity for for finite energy : (B.9).

Sufficiency for finite energy and that the transformation be bounded: (B.6) and $B < \infty$ in (B.8).

Sufficiency for a bounded inverse: (B.6) and $A > 0$ in (B.8).

Necessity and sufficiency for energy conservation : (B.11).

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1 AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE June 1991	3. REPORT TYPE AND DATES COVERED January–December 1990

4. TITLE AND SUBTITLE

THE DISCRETE WAVELET TRANSFORM

5 FUNDING NUMBERS

PE: 0601152N
PROJ: R00N0
SUBPROJ: 76-ZW51-01
ACC: DN300155

6. AUTHOR(S)

M. J. Shensa

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Ocean Systems Center
San Diego, CA 92152-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

NOSC TR 1426

9 SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of the Chief of Naval Research
Independent Research Programs (IR)
Arlington, VA 22217-5000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11 SUPPLEMENTARY NOTES

12a DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b DISTRIBUTION CODE

13 ABSTRACT *(Maximum 200 words)*

In a general sense, this report represents an effort to clarify the relationship of discrete and continuous wavelet transforms. More narrowly, it focuses on bringing together two separately motivated implementations of the wavelet transform, the *algorithme à trous* and Mallat's multiresolution decomposition. These algorithms are special cases of a single filter bank structure, the discrete wavelet transform, the behavior of which is governed by one's choice of filters. In fact, the *à trous* algorithm, originally devised as a computationally efficient implementation, is more properly viewed as a nonorthogonal multiresolution algorithm for which the discrete wavelet transform is exact. Moreover, we show that the commonly used Lagrange *à trous* filters are in one-to-one correspondence with the convolutional squares of the Daubechies filters for orthonormal wavelets of compact support.

A systematic framework for the discrete wavelet transform is provided, and conditions are derived under which it computes the continuous wavelet transform exactly. Suitable filter constraints for finite energy and boundedness of the discrete transform are also derived. Finally, relevant signal-processing parameters are examined and it is remarked that orthonormality is balanced by restrictions on resolution.

14 SUBJECT TERMS

wavelet transform	signal processing
filter banks	quadrature mirror filter
à trous algorithm	discrete wavelet transform

15 NUMBER OF PAGES

58

16 PRICE CODE

17 SECURITY CLASSIFICATION OF REPORT	18 SECURITY CLASSIFICATION OF THIS PAGE	19 SECURITY CLASSIFICATION OF ABSTRACT	20 LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAME AS REPORT

21a. NAME OF RESPONSIBLE INDIVIDUAL M. J. Shensa	21b. TELEPHONE *(Include Area Code)* (619) 553-5702	21c. OFFICE SYMBOL Code 632